

¶

# From Log4j to XZ

**Unsolvable Issues in the Software Supply Chain**

BSidesRedRocks

2024-11-15

Paul Novarese <pvn@huntedlabs.io>



 \$ whoami



**Paul Novarese**

**Hunted Labs**

**pvn@huntedlabs.io**

**Fediverse: @pvn@mas.to**



# ¶ Agenda

1. The Before Times
  2. Log4Shell
  3. SBOMs
  4. XZ Utils
- 
- Patch Faster is Broken (“dumpster fire”)
    - Most things don't even need to be fixed
  - The “Zero CVE” Goal is Wrong
    - “number go down”
  - What we Should Worry About
  - And what's not Important



# My Biases

- This talk is mainly about vulnerability management
  - (as opposed to regulatory compliance)
- My background is more Ops than Dev
- I empathize more with blue teams
- My day job is soaked in cloud native woo woo
- I have spent most of my career working in open source



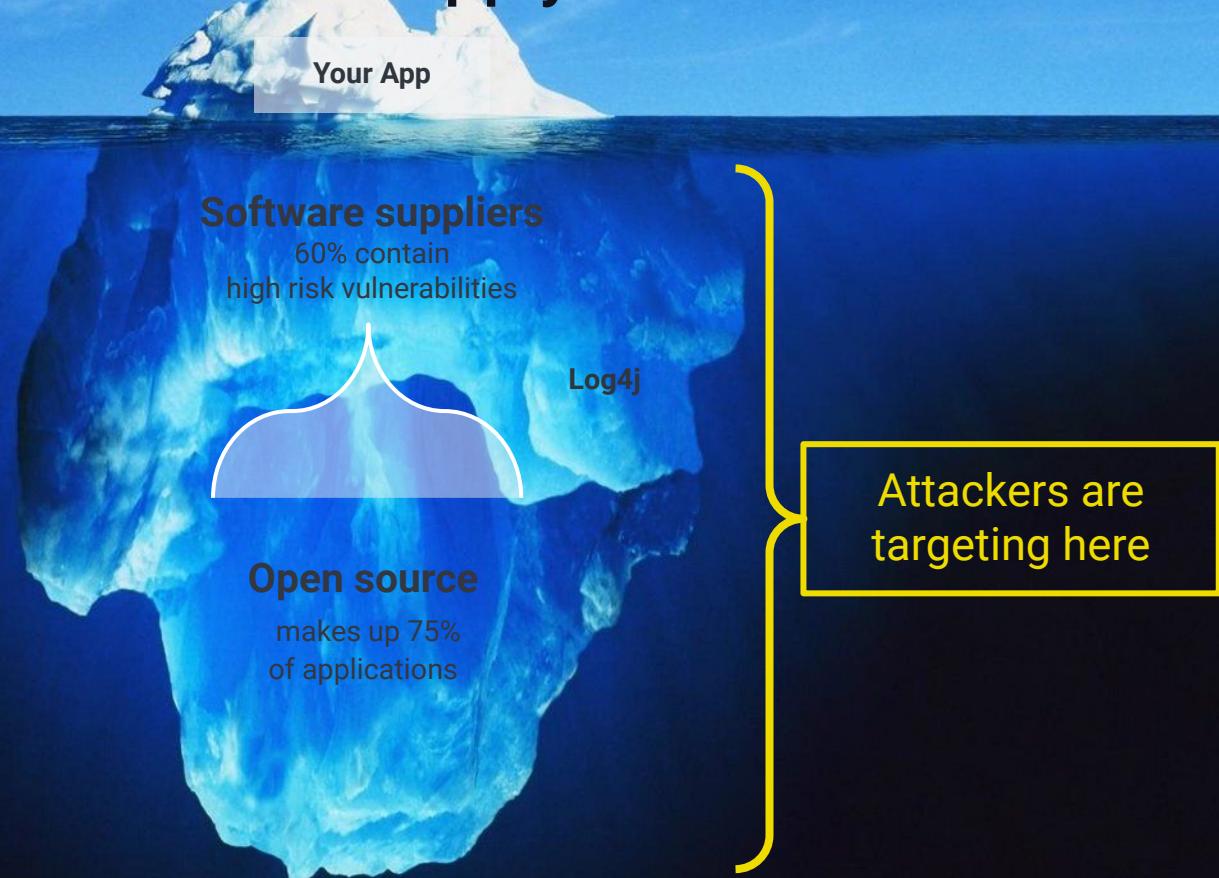
# The Before Times

Everything before Log4Shell



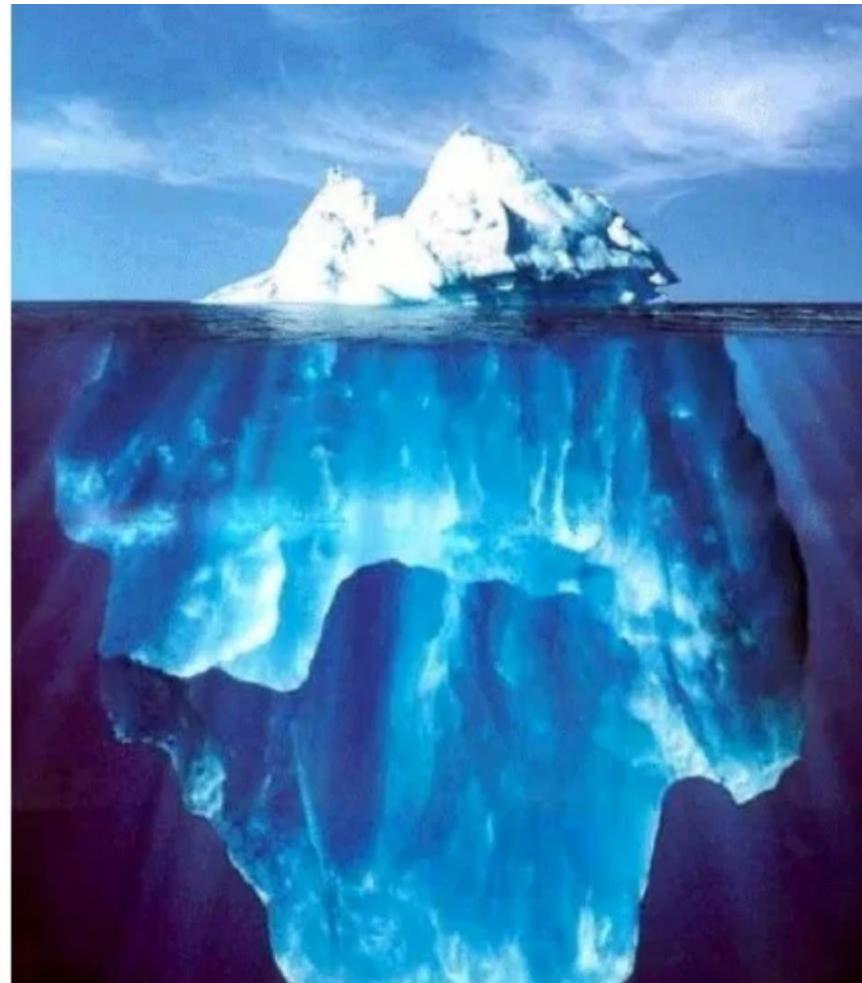
# Hidden Risk in the Software Supply Chain

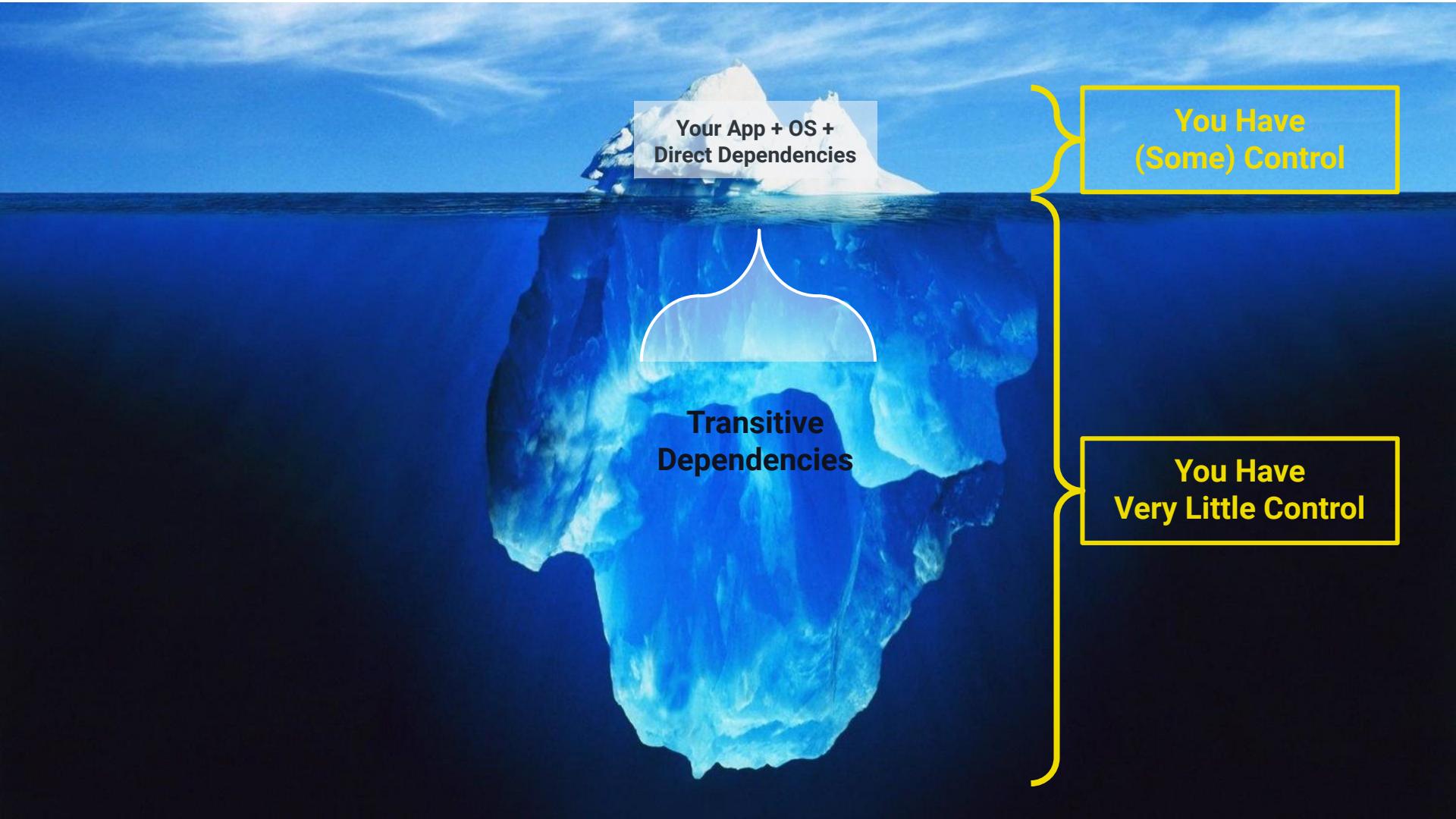
## Risk in the Software Supply Chain



# Free is Just the Tip of the Iceberg: Open Source Library System Software

Lori Bowen Ayre  
[lori.ayre@galecia.com](mailto:lori.ayre@galecia.com)  
METRO Webinar  
October 6, 2009



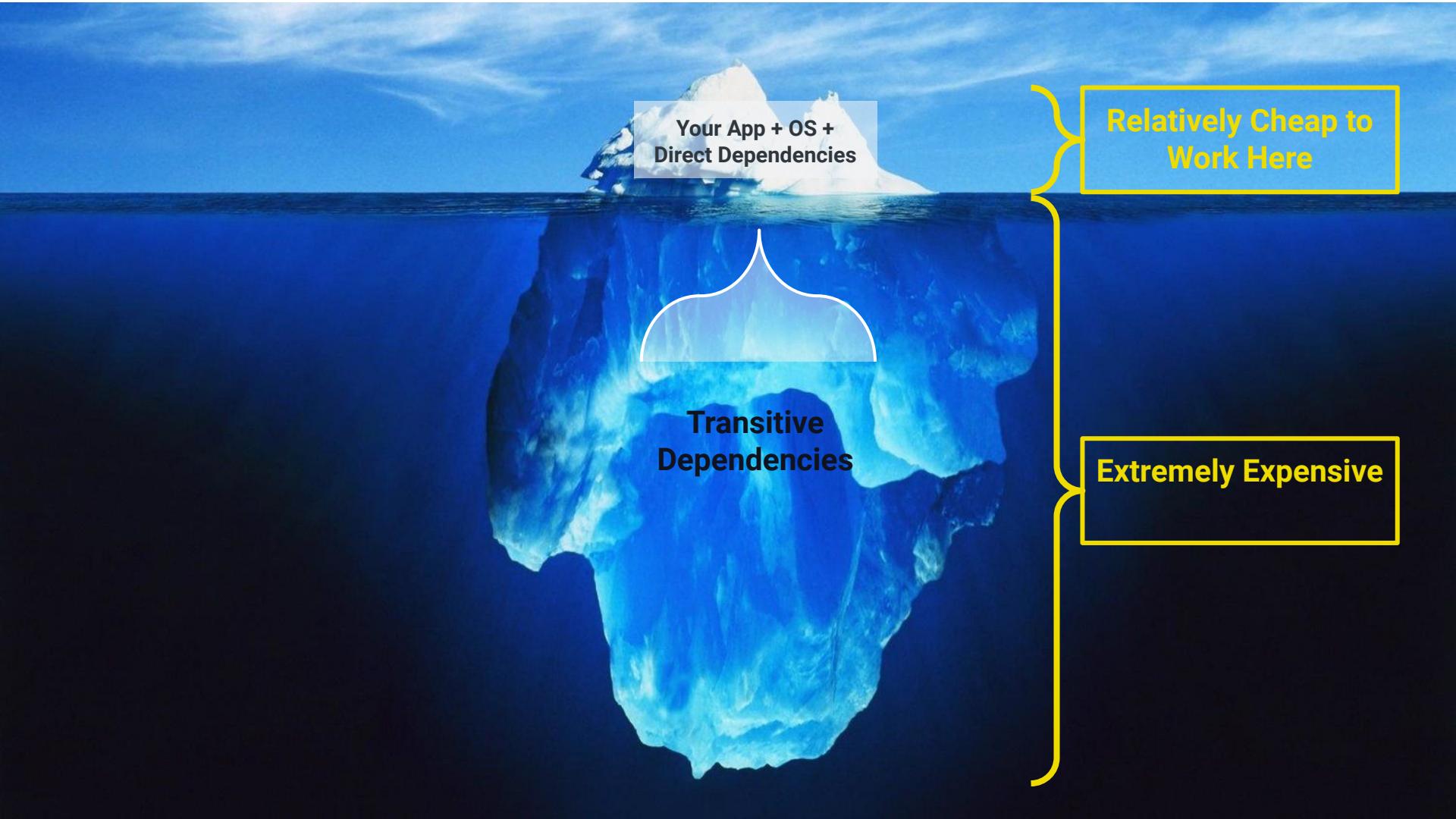


Your App + OS +  
Direct Dependencies

Transitive  
Dependencies

You Have  
(Some) Control

You Have  
Very Little Control



Your App + OS +  
Direct Dependencies

Transitive  
Dependencies

Relatively Cheap to  
Work Here

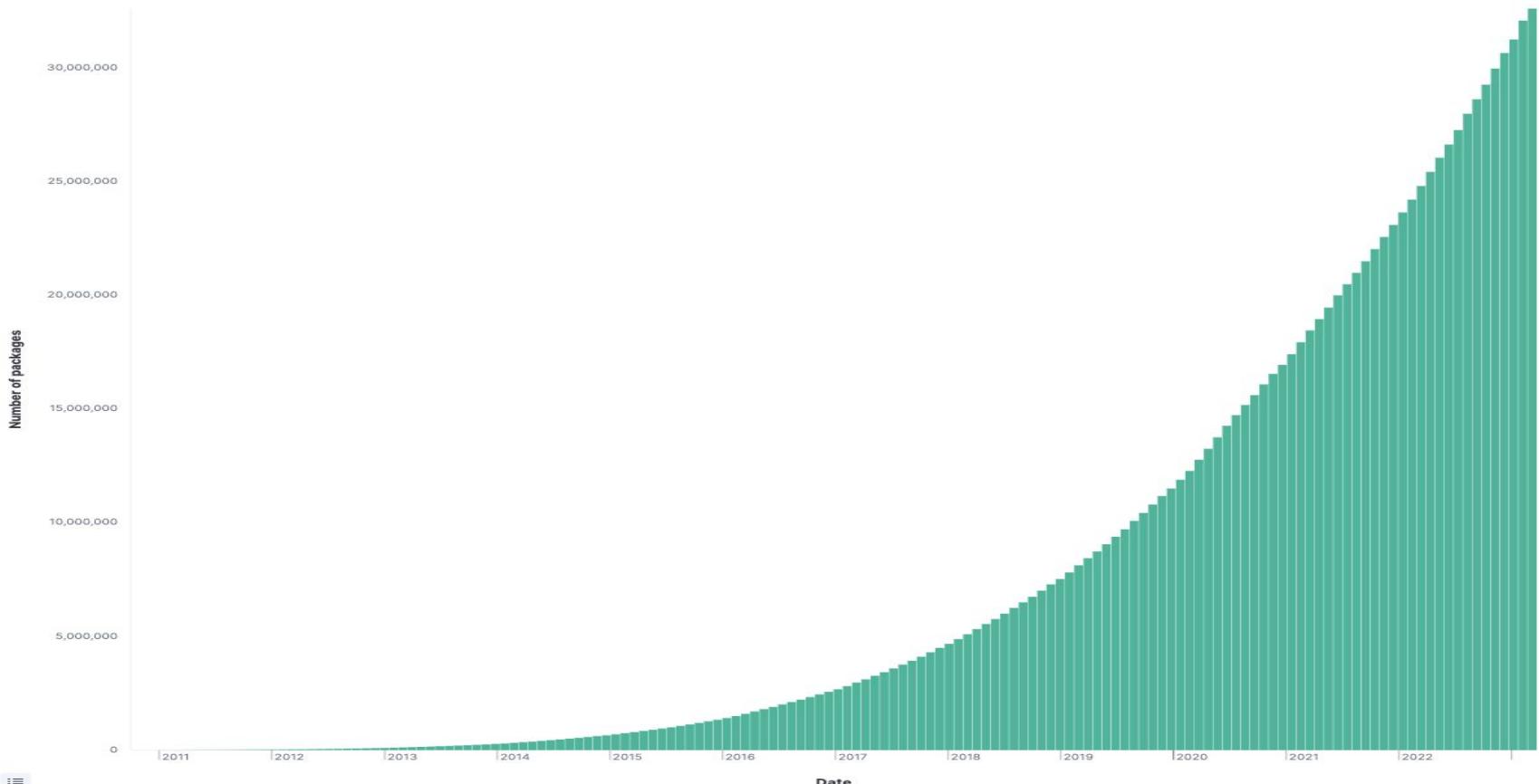
Extremely Expensive



# Notes on this Metaphor

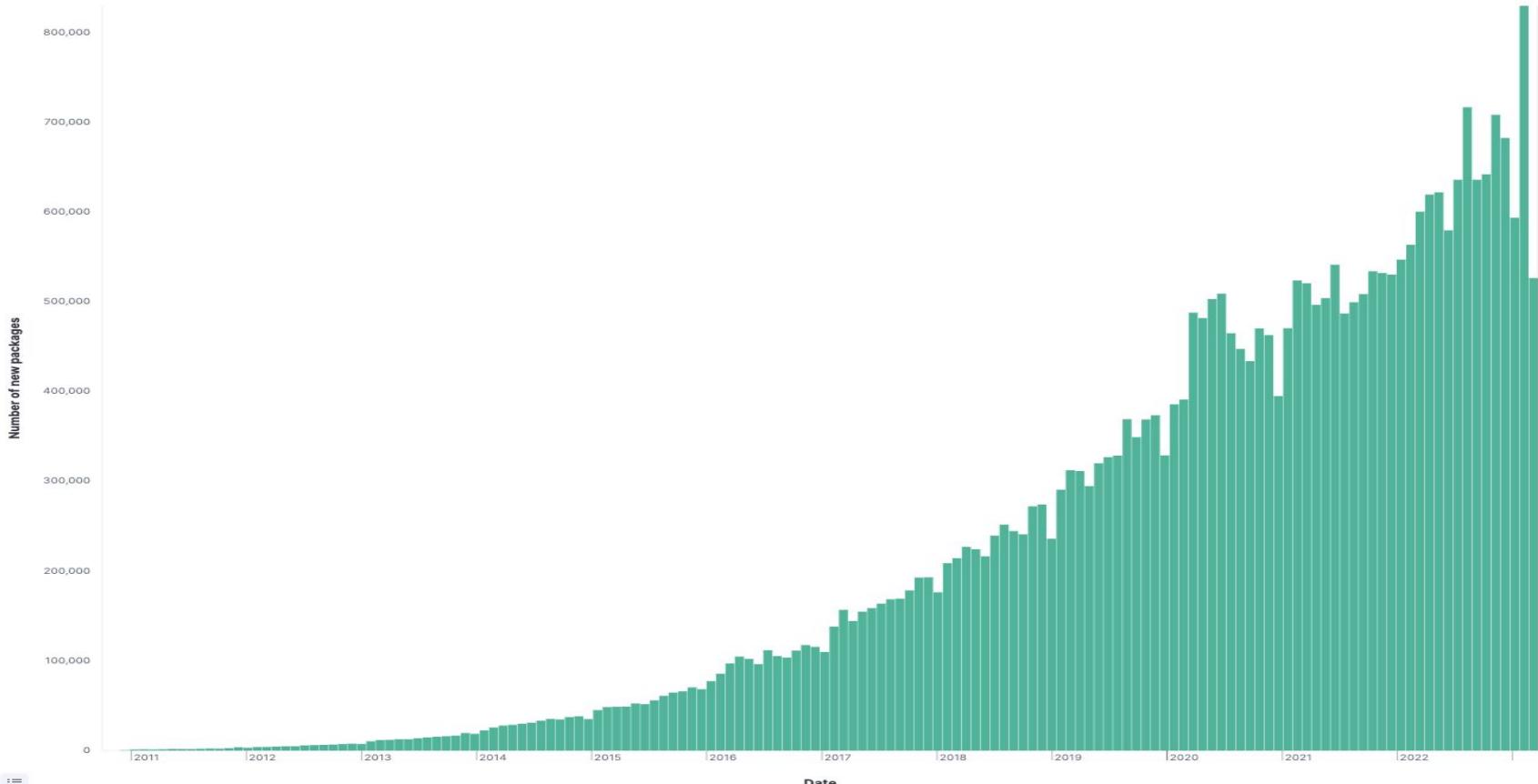
- You've seen this iceberg metaphor. I've used this metaphor 100 times, I've criticized this metaphor.
- This is an OLD metaphor
- Things have changed a lot but we're still thinking about old systems
- <https://www.slideshare.net/loriayre/open-source-library-system-software-free-is-just-the-tip-of-the-iceberg>
- They're attacking the bottom now - that's a supply chain attack
- But really, the top isn't "your code" - the top is your direct dependencies, bottom is transitive
- You can only directly control what's at the top
- They're attacking the whole iceberg, but you probably only know about the stuff at the top
- The change is largely due to the massive rise in software package managers
- The CVE system predates this change and hasn't really evolved

# ¶ Open Source is Bigger Than You Can Imagine

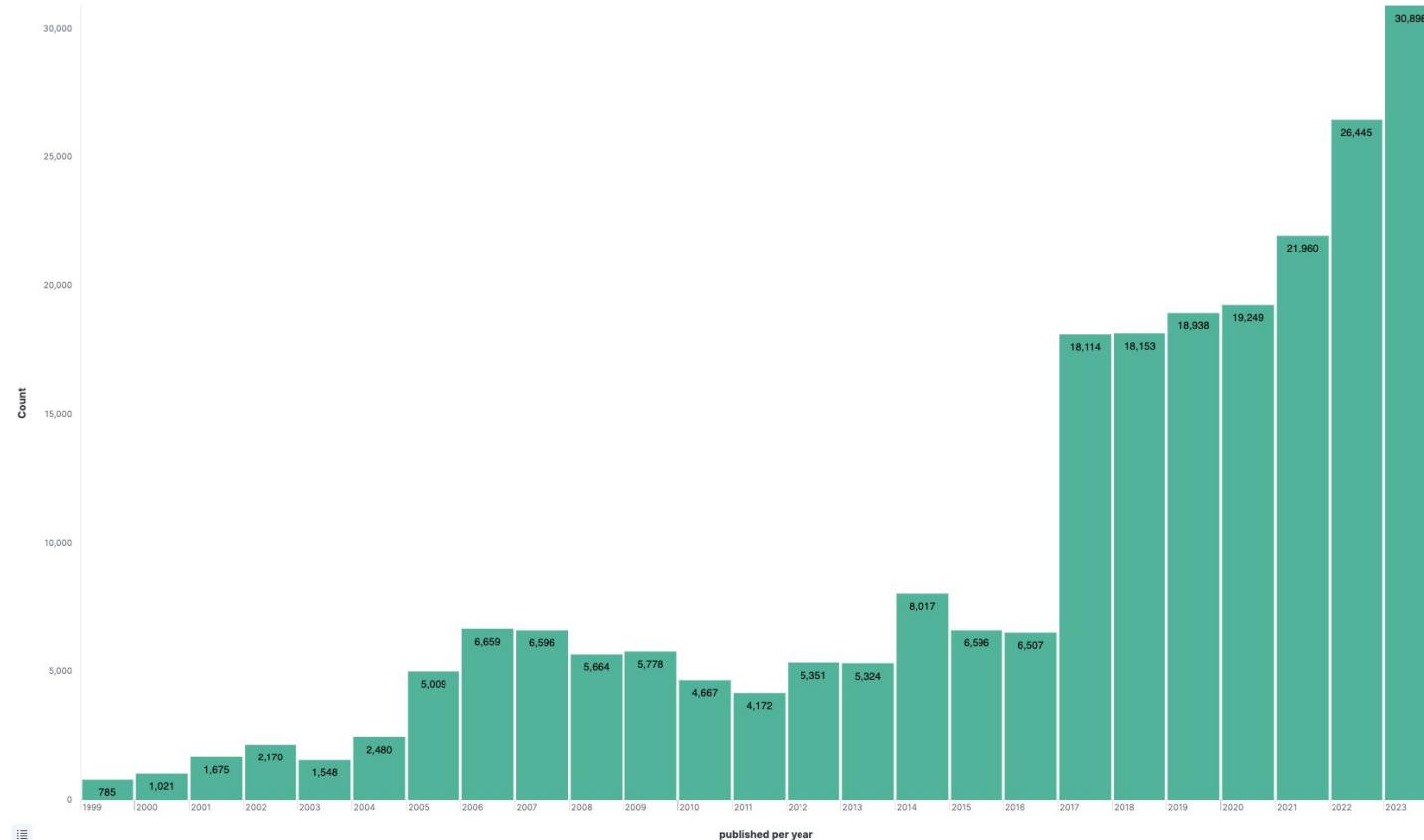




# Open Source is Bigger Than You Can Imagine



# And CVE Growth



# ¶ Open Source is Huge

- NPM introduced 2010
- 43 million packages (as of April)
- Approx 1,000,000 new packages \*\*per month\*\*
- That's just NPM!

[npmjs.org](https://npmjs.org)

3,732,919 packages

42,958,444 versions

850,084 maintainers

231,488 namespaces

752,313 keywords

256,314,168,001 downloads



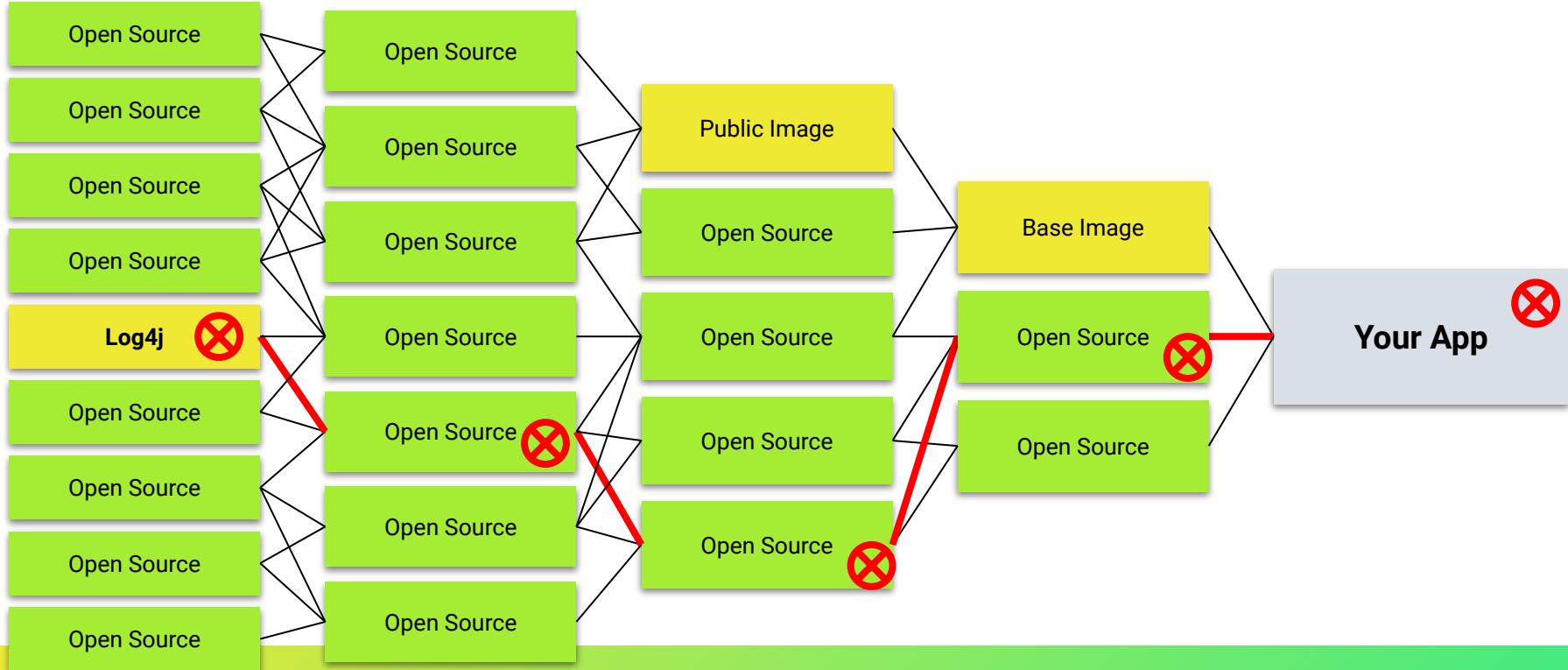


# Log4Shell

The Awakening

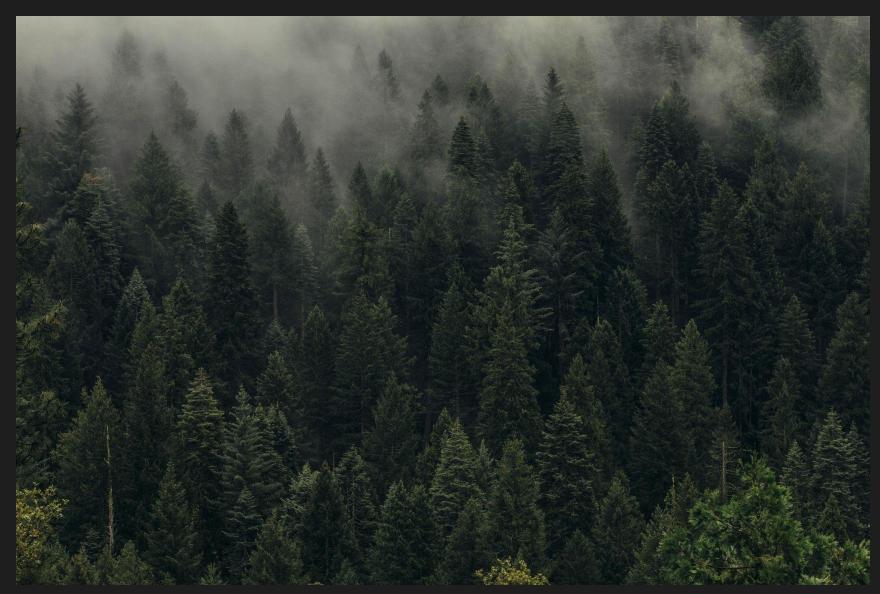


# The Software Supply Chain



# ¶ Stop Thinking About Open Source like a Vendor

This



Not this



# Open Source Software Supply Chains

- Red Hat **IS** a supplier (if you are paying them)
  - they assume responsibility in exchange for money
- npm is **NOT** a supplier
- A lot of critical plumbing is maintained by unpaid guys who have day jobs, take vacations, etc.

# Log4Shell is Radioactive and Immortal

CYBERSECURITY ADVISORY

## 2023 Top Routinely Exploited Vulnerabilities

Release Date: November 12, 2024

Alert Code: AA24-317A

- [CVE-2021-44228](#): This vulnerability, known as Log4Shell, affects Apache's Log4j library, an open source logging framework incorporated into thousands of products worldwide.
  - Allows the execution of arbitrary code.
    - An actor can exploit this vulnerability by submitting a specially crafted request to a vulnerable system, causing the execution of arbitrary code.
    - The request allows a cyber actor to take full control of a system.
    - The actor can then steal information, launch ransomware, or conduct other malicious activity.
    - Malicious cyber actors began exploiting the vulnerability after it was publicly disclosed in December 2021.

# Log4Shell Recap

- Growth of dependencies is now obvious
- The term “software supply chain” starts to show up
- Many people hear about SBOMs for the first time
- STILL to this day on CISA’s Top 15 Frequently Exploited List
  - (Also the only open source component on that list)
  - **We’ll be dealing with this for DECADES**
    - <https://www.cisa.gov/sites/default/files/2024-11/aa24-317a-2023-top-routinely-exploited-vulnerabilities.pdf>



# SBOMs

A brief note



# What is an SBOM?



# If We Knew What We are Consuming

- People spent insane amounts of time just finding log4j, because nobody knew where (or even if) it was hiding
- Knowing = Faster Remediation
- SBOMs help, a LOT, but... “a phone book is not illuminating”
  - They aren’t a silver bullet
  - Scanners aren’t perfect (e.g. can’t penetrate binary blobs, cf. OpenSSL3.)
  - Not all SBOMs are equal
  - SBOMs aren’t ubiquitous (yet) (producers aren’t reliably supplying them)
  - SBOMs are more accurate and useful when producers/maintainers generate them BUT something is better than nothing
  - SBOM management is hard
  - Any SBOM generated before an actual build is suspect (transitive deps)
  - SBOM Everywhere: <https://github.com/ossf/sbom-everywhere>



# XZ and Beyond

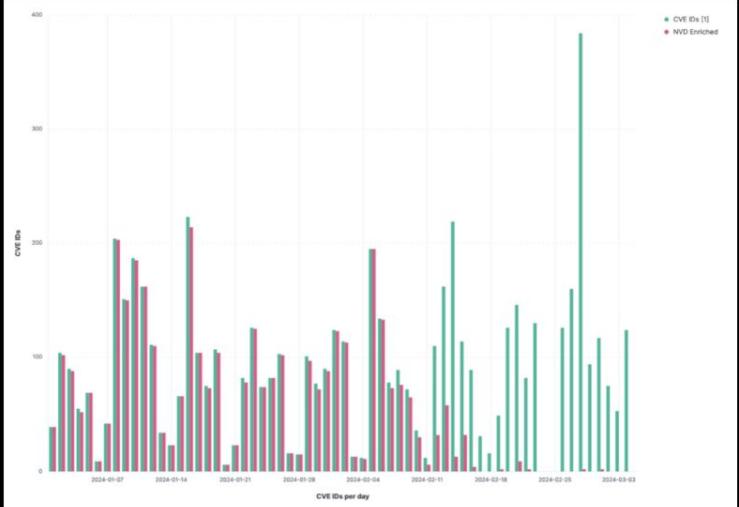
An Amazing Thing Happened at a Unique Moment in Time

# NATIONAL VULNERABILITY DATABASE



## NOTICE

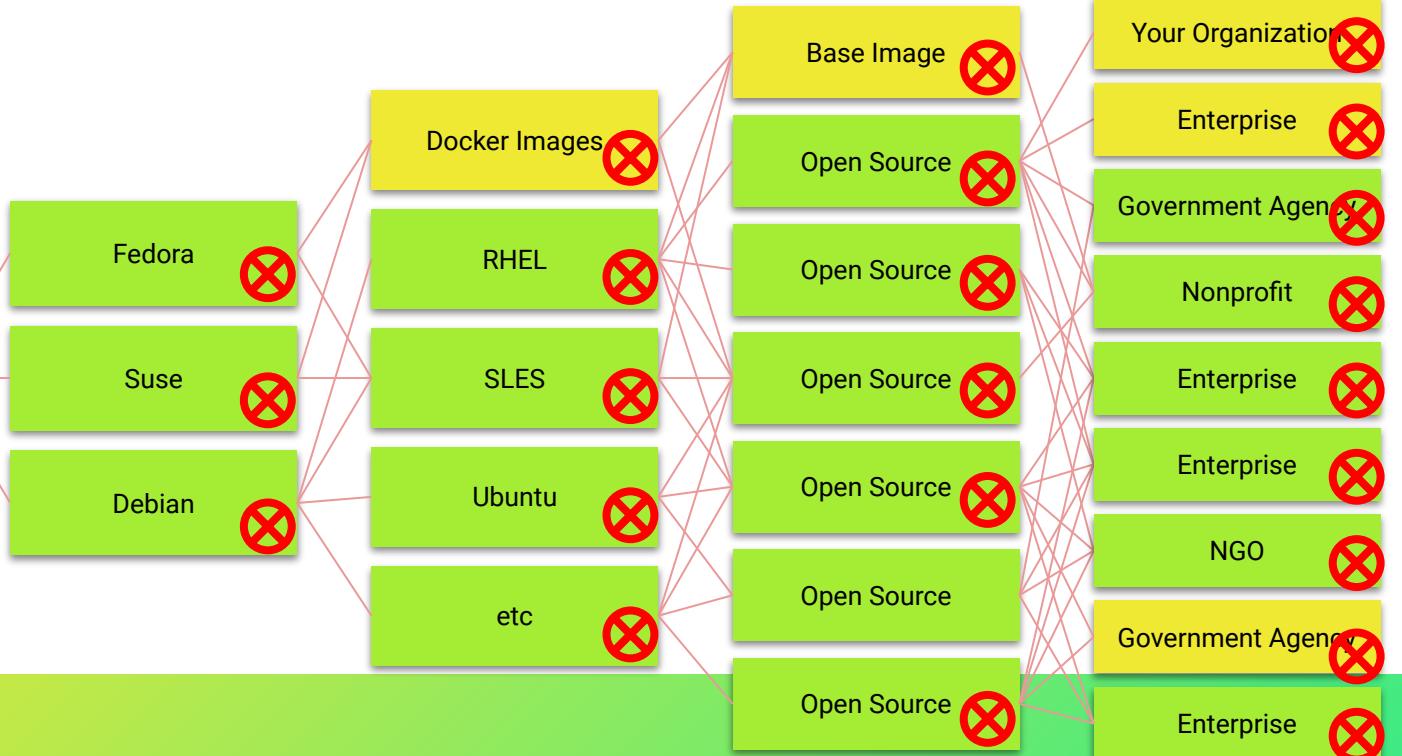
NIST is currently working to establish a consortium to address challenges in the NVD program and develop improved tools and methods. You will temporarily see delays in analysis efforts during this transition. We apologize for the inconvenience and ask for your patience as we work to improve the NVD program.



# The Jia Tan Reverse Funnel Plan



XZ





# The Takeaways



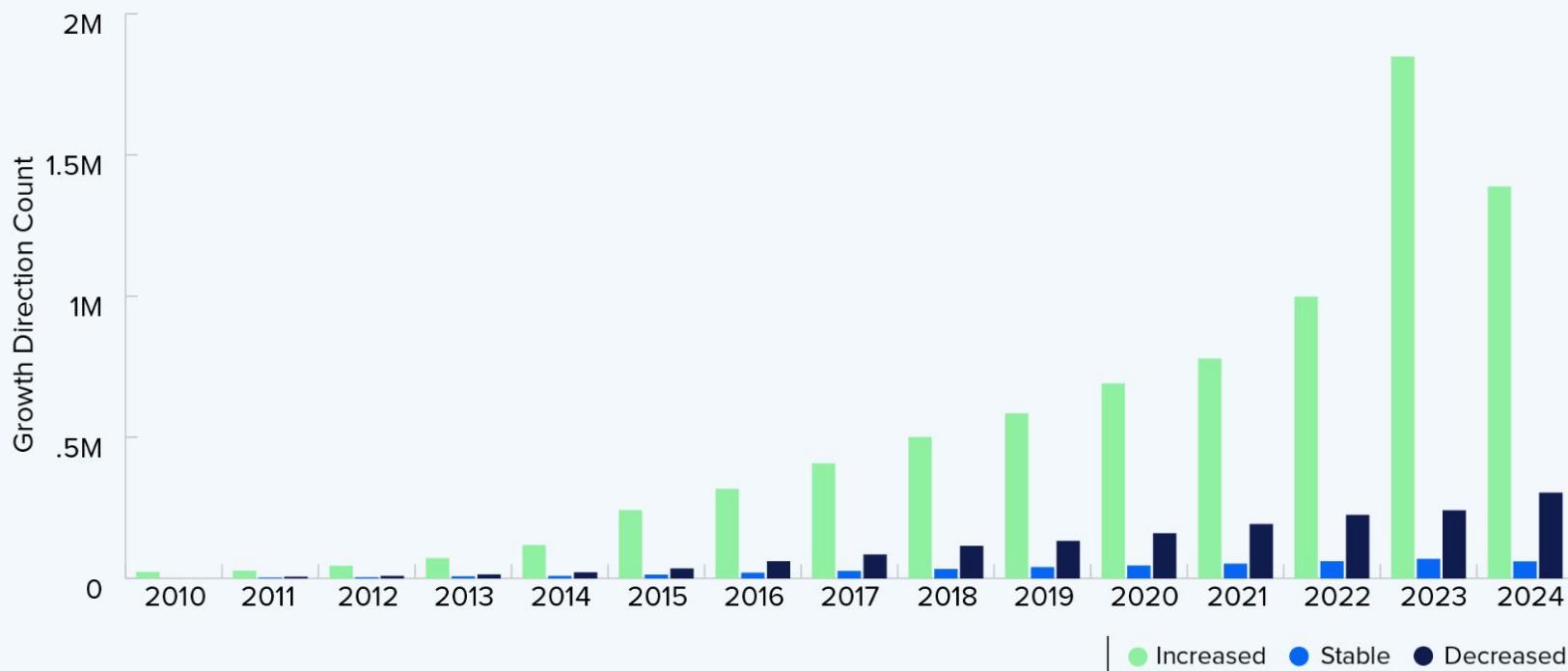
Open source is different

**There's nothing wrong with open source,  
this is how it works**

**There's something wrong with  
what we expect from open source**

**FIGURE 1.2**

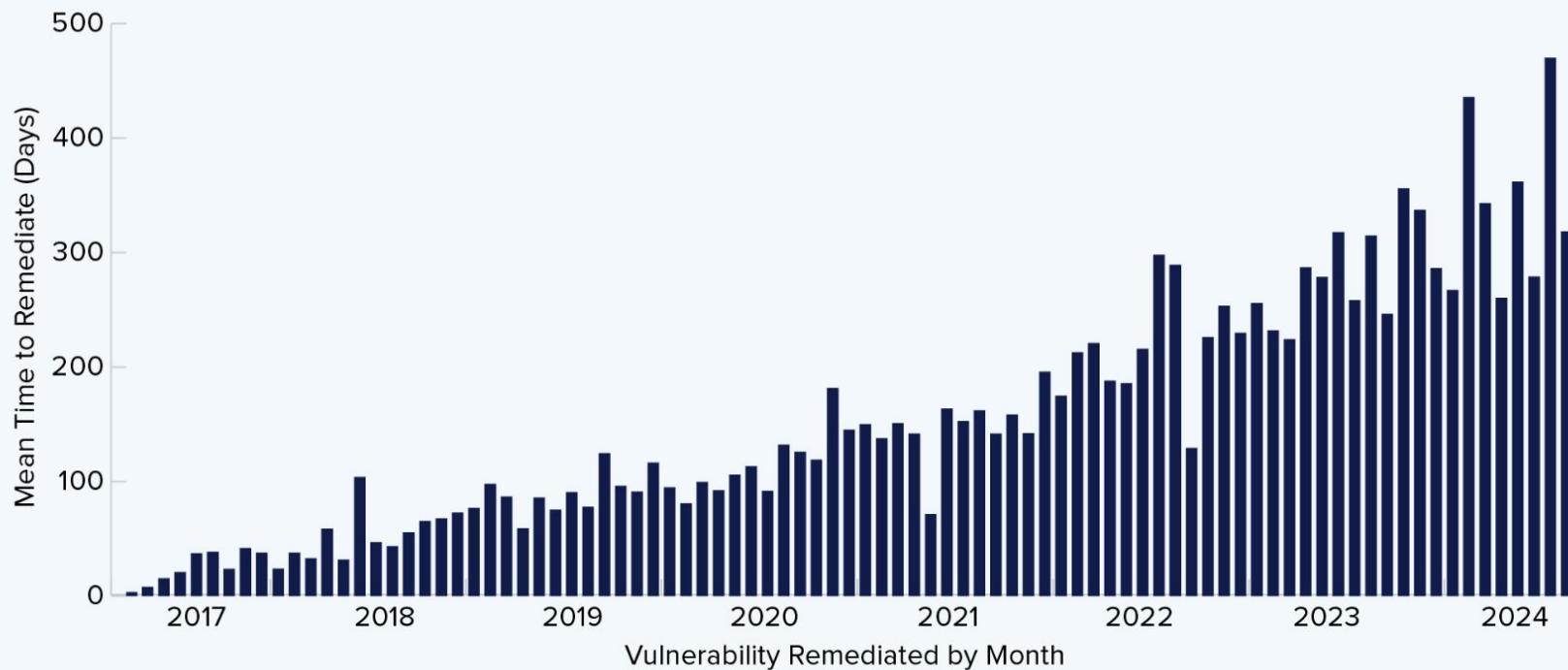
## Release Frequency of Open Source Projects



Projects that released faster, slower or the same as the prior year.

**FIGURE 1.3**

## Rate of Vulnerability Remediation Over Time



How long a project took to remediate known vulnerabilities in their dependencies.



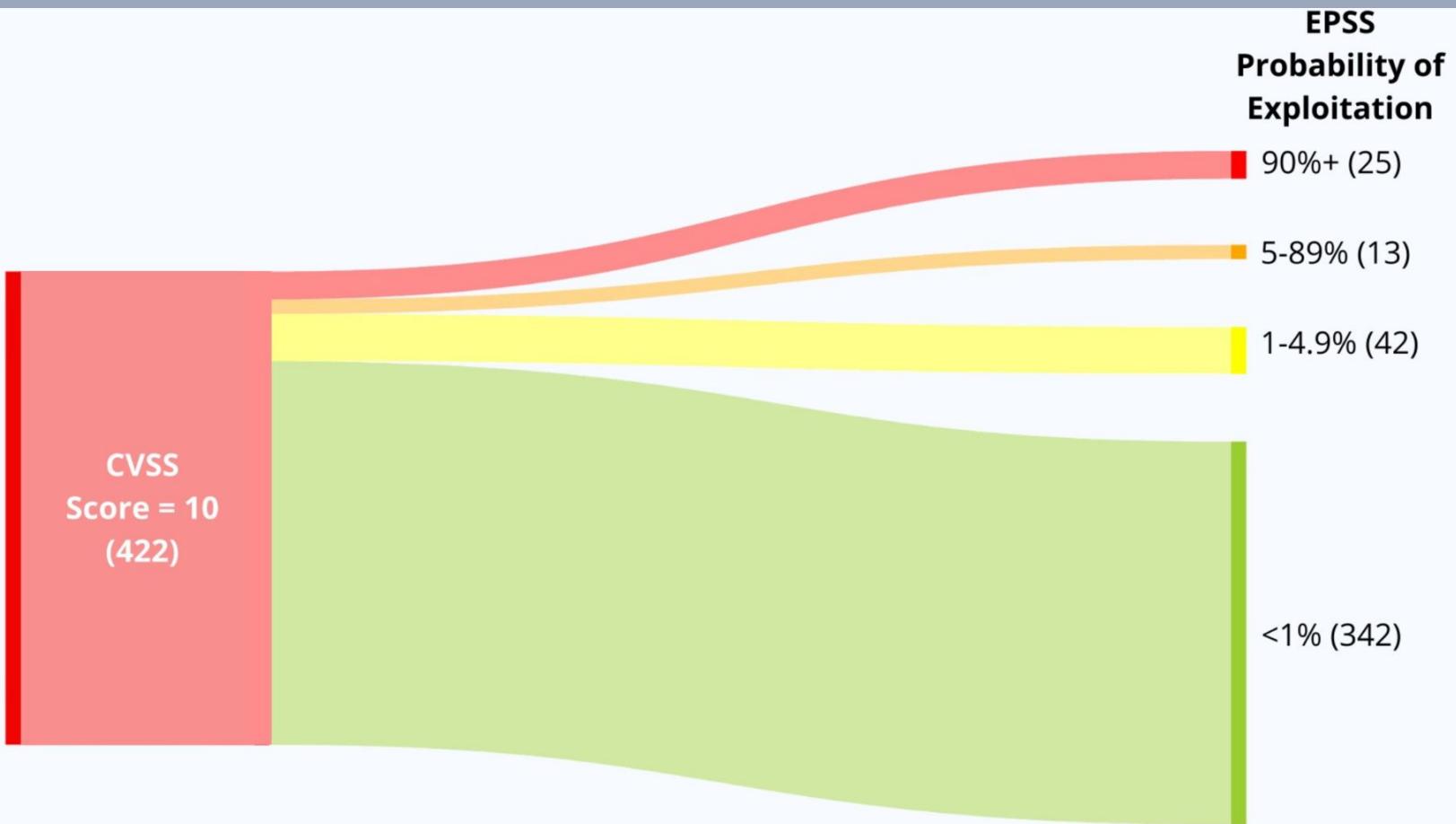






# Takeaway: Patch Faster is Broken and Chasing CVE 0 is a Losing Battle

- CVEs increasing faster than they can be fixed
- Most of these are not important anyway
- But they all have CVSS scores > 9.8 so you can't tell which ones ARE important
- GHSAs (more transparent than CVEs)
- **CISA KEV + EPSS**
- VEX, CSAF, OpenSSF Malicious Packages Repository are helpful
- GitHub Insights and other project health metrics
  - This is (currently) a very manual process
  - But it's getting a lot easier





# Takeaway: Open Source Project Health/Insights

- This is PROACTIVE (better advisory data, etc is about reactive improvements)
- This is (currently) a manual process (getting easier)
- Evaluating project health isn't directly about safety, it's about all keeping track of all those deps in the iceberg
- **Are the projects you're depending on healthy, will you be able to work with them?**
- **Ransomware attacks and the software supply chain as a vector are peanut butter and chocolate**

- Pulse
- Contributors
- Community Standards
- Commits
- Code frequency
- Dependency graph
- Network
- Forks
- Actions Usage Metrics
- Actions Performance Metrics

November 4, 2024 – November 11, 2024

Period: 1 week ▾

## Overview

13 Active pull requests

6 Active issues

4 Merged pull requests

9 Open pull requests

1 Closed issue

5 New issues

Excluding merges, 5 authors have pushed 4 commits to main and 12 commits to all branches. On main, 3 files have changed and there have been 13 additions and 1 deletions.



4 Pull requests merged by 4 people

## Restore log on UI teardown

#3427 merged 3 days ago

## doc: Add official Syft logo license information

#3421 merged 4 days ago

## chore(deps): bump anchore/sbom-action from 0.17.6 to 0.17.7

#3418 merged 5 days ago

## chore: build release sbom from go.mod

#3417 merged last week

9 Pull requests opened by 4 people

## update node classifier to support 6.x

#3419 opened 5 days ago

## Support scanning files in mount namespaces

#3423 opened 4 days ago

## chore(deps): update stereoscope to 120d9ea511e2f7a9887b443c52e66cd19bb80b43

#3424 opened 4 days ago

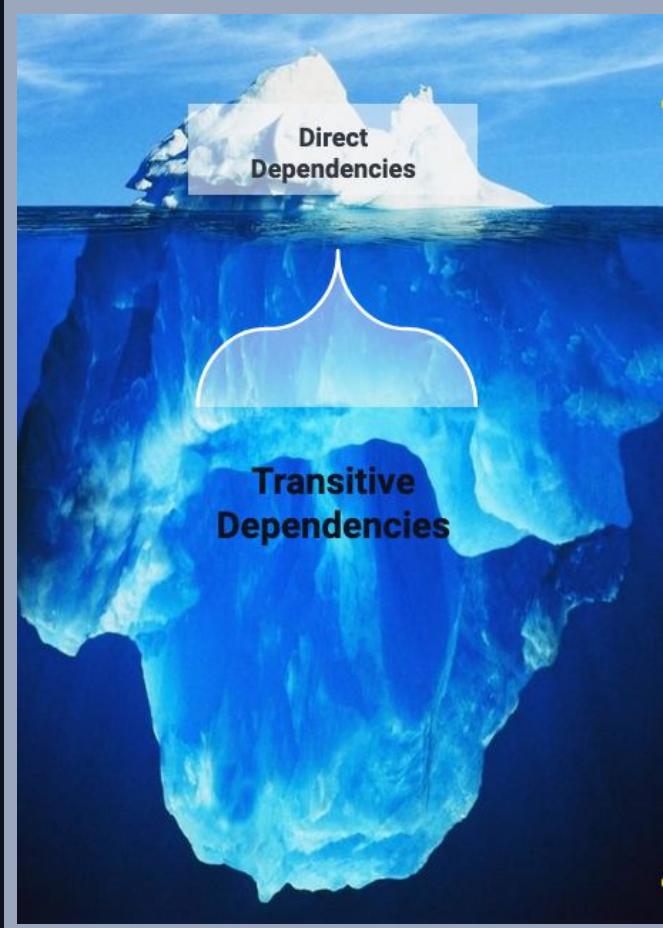
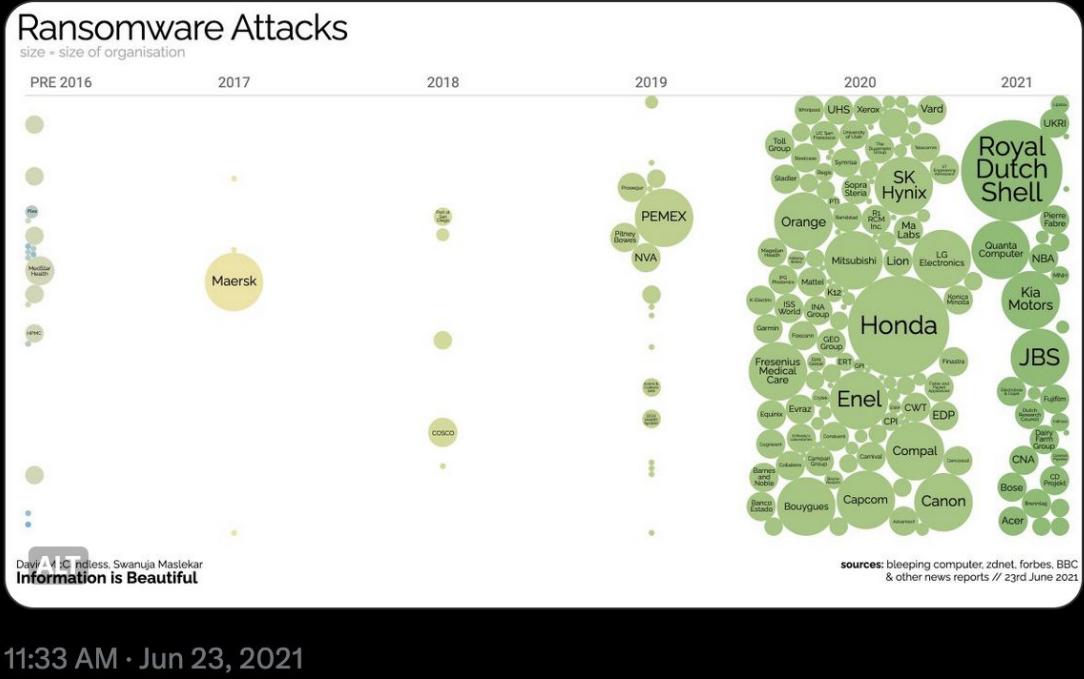
```
"purl": "pkg:gem/zlib@2.0.0",
"metadataType": "GemMetadata",
"metadata": {
  "name": "zlib",
  "version": "2.0.0",
  "files": [
    "ext/zlib/extconf.rb",
    "zlib.so"
  ],
  "authors": [
    "Yukihiro Matsumoto",
    "UENO Katsuhiko"
  ],
  "homepage": "https://github.com/ruby/zlib"
}
```



**Information is Beautiful**  
@infobeautiful

Are #Ransomware attacks increasing? I think #Ransomware attacks are increasing...

interactive: [bit.ly/3h1IYPs](https://bit.ly/3h1IYPs)



# THE PROBLEM

## Comments

iso27032

Photo

iso27032 - 3 days ago



Bill,

This is not the first time the software supply chain has seen mischief. Would you recommend that all those who create and maintain packages should be properly registered and vetted?

Thanks.

# THE “SOLUTION”



A woman with long blonde hair, wearing a purple top, is looking directly at the camera with a neutral expression. She is holding a white lightbulb in her right hand, which is positioned near her face. The background is dark and out of focus.

**STOP TRYING TO MAKE**

**verified  
open source  
contributor**

**HAPPEN**

**ITS NOT GOING TO HAPPEN!**



## Takeaway: Open Source Will Not Work in the Authoritarian Dystopia

- Registration/Authentication/Verification will backfire
- Many projects will move to weird places
- A lot of people will just stop contributing

H

**HUNTER**  
LABS





# Footnotes

- Sonatype: State of the Software Supply Chain  
<https://www.sonatype.com/state-of-the-software-supply-chain/introduction>
- Tidelift: State of the Open Source Maintainer  
<https://explore.tidelift.com/2024-survey/2024-tidelift-state-of-the-open-source-maintainer-report>
- Anchore: Software Supply Chain Security Report  
<https://get.anchore.com/2024-software-supply-chain-security-report/>
- Thomas Depierre: I am not a Supplier  
<https://www.softwaremaxims.com/blog/not-a-supplier>
- The Double-Edged Sword of Increased Vulnerability Data  
<https://github.blog/security/supply-chain-security/securing-the-open-source-supply-chain-the-essential-role-of-cves/>
- Open Source is Bigger Than You Can Imagine  
<https://anchore.com/blog/open-source-is-bigger-than-you-imagine/>
- 2023 Top Routinely Exploited Vulnerabilities  
<https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-317a>
- Patrick's CVE Diagrams  
[https://www.linkedin.com/posts/patrickmgarrity\\_the-evolution-of-patricks-sankey-matics-activity-7118334146728357888-zxxn/](https://www.linkedin.com/posts/patrickmgarrity_the-evolution-of-patricks-sankey-matics-activity-7118334146728357888-zxxn/)
- possible origin of the iceberg  
<https://www.slideshare.net/loriayre/open-source-library-system-software-free-is-just-the-tip-of-the-iceberg>
- Log4Shell logo: [https://en.wikipedia.org/wiki/File:Log4Shell\\_logo.png](https://en.wikipedia.org/wiki/File:Log4Shell_logo.png)
- xz logo: <https://infosec.exchange/@jerry/112186387514069376>

# XZ Reading List

Technologist vs spy: the XZ backdoor debate

<https://lcamtuf.substack.com/p/technologist-vs-spy-the-xz-backdoor>

General XZ roundups

<https://boehs.org/node/everything-i-know-about-the-xz-backdoor>

<https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/>

FAQ on the XZ compromise/backdoor CVE-2024-3094

<https://gist.github.com/thesamesam/223949d5a074ebc3dce9ee78baad9e27>

examination of claims of technical solutions to XZ and why they're wrong

<https://federated.saagarjha.com/notice/AgPahhBPr9xHXMpWi>

OSS backdoors: the folly of the easy fix

<https://lcamtuf.substack.com/p/oss-backdoors-the-allure-of-the-easy>

deep inspection of the backdoor injection

<https://research.swtch.com/xz-script>

<https://gynvael.coldwind.pl/?lang=en&id=782>

interactions in open source projects (examination of XZ infiltration)

<https://robmenschling.com/blog/posts/2024/03/30/a-microcosm-of-the-interactions-in-open-source-projects/>

thread from november 2023 theorizing about a long con threat actor assuming control of a major project

<https://infosec.exchange/@mariuxdeangelo/111348817163534252>

thread exploring pressure on XZ maintainer to hand off control of the project

<https://twitter.com/robmen/status/1774067844785086775>

bullying as a vulnerability in open source

<https://www.404media.co/xz-backdoor-bullying-in-open-source-software-is-a-massive-security-vulnerability>

tracking Jia Tan's commit timestamps

<https://twitter.com/birchb0y/status/1773871381890924872>

examining Jia Tan's complete github commit history

<https://huntedlabs.com/where-the-wild-things-are-a-complete-analysis-of-jiat95-github-history>

looking into the "Jia Tan" persona

<https://www.wired.com/story/jia-tan-xz-backdoor/>

Sloppy OpenSSF statement (later redacted) implying Scorecard indicated XZ issues

<https://web.archive.org/web/20240331024907/https://openssf.org/blog/2024/03/30/xz-backdoor-cve-2024-3094/>

Lessons from XZ Utils: Achieving a More Sustainable Open Source Ecosystem

<https://www.cisa.gov/news-events/news/lessons-xz-utils-achieving-more-sustainable-open-source-ecosystem>

# Log4Shell Reading List

Dealing with log4shell (detection, mitigation, workarounds)

<https://cloudsecurityalliance.org/blog/2021/12/14/dealing-with-log4shell-aka-cve-2021-44228-aka-the-log4j-version-2/>

Keeping up with log4shell (post mortem)

<https://cloudsecurityalliance.org/blog/2021/12/16/keeping-up-with-log4shell-aka-cve-2021-44228-aka-the-log4j-version-2/>

Mysterious tweet hinting at the exploit

<https://twitter.com/sirifu4k1/status/1468951859381485573>

Another mysterious tweet:

<https://twitter.com/CattusGlavo/status/1469010118163374089>

“THE” pull request:

<https://github.com/apache/logging-log4j2/pull/608>

Cloudflare digs for evidence of pre-disclosure exploits in the wild:

<https://twitter.com/eastdakota/status/1469800951351427073>

# Open Source Reconnaissance Reading List

NPM Provenance: The Missing Security Layer in Popular JavaScript Libraries

<https://medium.com/exaforce/npm-provenance-the-missing-security-layer-in-popular-javascript-libraries-b50107927008>

Your dependencies have dependencies: new features to assess risk

<https://dev.to/stacklok/your-dependencies-have-dependencies-new-features-to-assess-risk-3f1b>

Repo Swatting

<https://www.bsidesmelbourne.com/2024-repo.html>

<https://github.com/6mile/repo-swatting> (hopefully slides will be posted soon)

Securing open source software: Whose job is it, anyway?

[https://www.theregister.com/2024/03/08/securing\\_opensource\\_software\\_whose\\_job/](https://www.theregister.com/2024/03/08/securing_opensource_software_whose_job/)

Maltego Cyber Investigation Platform &c

<https://www.maltego.com/>

The US Federal Government Understands that open source is not a supplier

<https://www.linkedin.com/feed/update/urn:li:activity:7073021512030511104/>

identifying vulnerabilities in open source codebases at scale

<https://github.com/chebuya/SASTsweep>

## Bad Ideas Around Enforced Contributor Identity and Authentication:

Malicious PyPI package with 37,000 downloads steals AWS keys

<https://www.bleepingcomputer.com/news/security/malicious-pypi-package-with-37-000-downloads-steals-aws-keys/>

LLM Code Authorship Detection (this is a bad idea and will probably make things worse)

<https://apiiro.com/blog/llm-code-author-detection-unmasking-malicious-package-contributions/>

Digital Identity Attestation Roundup

<https://openssf.org/blog/2021/01/27/digital-identity-attestation-roundup/>

Building Trust Within Open Source Software

<https://www.identity.com/building-trust-within-open-source-software/>

## This isn't a problem specific to Open Source:

North Korean hacker got hired by US security vendor, immediately loaded malware

<https://arstechnica.com/tech-policy/2024/07/us-security-firm-unwittingly-hired-apparent-nation-state-hacker-from-north-korea/>

Twitter employee is convicted in Saudi spy case

<https://www.cnn.com/2022/08/09/tech/former-twitter-employee-conviction/index.html>

# Projects and Data Sources

NPM Provenance: The Missing Security Layer in Popular JavaScript Libraries

<https://medium.com/exaforce/npm-provenance-the-missing-security-layer-in-popular-javascript-libraries-b50107927008>

Your dependencies have dependencies: new features to assess risk

<https://dev.to/stacklok/your-dependencies-have-dependencies-new-features-to-assess-risk-3f1b>

Repo Swatting

<https://www.bsidesmelbourne.com/2024-repo.html>

<https://github.com/6mile/repo-swatting> (hopefully slides will be posted soon)

Securing open source software: Whose job is it, anyway?

[https://www.theregister.com/2024/03/08/securing\\_opensource\\_software\\_whose\\_job/](https://www.theregister.com/2024/03/08/securing_opensource_software_whose_job/)

Maltego Cyber Investigation Platform &c

<https://www.maltego.com/>

The US Federal Government Understands that open source is not a supplier

<https://www.linkedin.com/feed/update/urn:li:activity:7073021512030511104/>

identifying vulnerabilities in open source codebases at scale

<https://github.com/chebuya/SASTsweep>

OpenSSF Malicious Packages Repository:

<https://openssf.org/blog/2023/10/12/introducing-openssfs-malicious-packages-repository/>

Common Security Advisory Framework

<https://oasis-open.github.io/csaf-documentation/>

Exploit Prediction Scoring System

<https://www.first.org/epss/>

CISA Known Exploited Vulnerability Catalog

<https://www.cisa.gov/known-exploited-vulnerabilities-catalog>

Vulnerability Exploitability Exchange

<https://cyclonedx.org/capabilities/vex/>

GitHub Advisory Database

<https://github.com/advisories>

GitHub Insights

<https://docs.github.com/en/issues/planning-and-tracking-with-projects/viewing-insights-from-your-project/about-insights-for-projects>

Open Source Insights

<https://deps.dev/>



# CVE/NVD Brokenness Reading List

Filling the NVD data gap

<https://github.com/anchore/nvd-data-overrides>

NVD Chaos Podcast

<https://resiliencycyber.substack.com/p/s6e11-josh-bressers-and-dan-lorenc>

Identifying Software

<https://quix.gnu.org/en/blog/2024/identifying-software/>

CVEs CWEs CVSS and It's Discontents

<https://www.linkedin.com/pulse/cves-cwes-cvss-its-discontents-sherif-mansour>

Open Source Security Podcast Episode 392 – Curl and the calamity of CVE

<https://openourcesecurity.io/2023/09/10/episode-392-curl-and-the-calamity-of-cve/>

Shedding Light on CVSS Scoring Inconsistencies

<https://arxiv.org/abs/2308.15259>

My previous DevOpsDays 2022 talk (Learn From Log4Shell):

[https://www.youtube.com/watch?v=PiNtIL\\_oN0k](https://www.youtube.com/watch?v=PiNtIL_oN0k)

<https://github.com/pnovarese/2022-devopsdays>

Probably Don't Rely on EPSS Yet:

<https://insights.sei.cmu.edu/blog/probably-dont-rely-on-epss-yet/>

CVE-2020-19909 is everything that is wrong with CVEs:

<https://daniel.haxx.se/blog/2023/08/26/cve-2020-19909-is-everything-that-is-wrong-with-cves/>

# L Software Supply Chains Reading List

Hackers poison source code from largest Discord bot platform

<https://www.bleepingcomputer.com/news/security/hackers-poison-source-code-from-largest-discord-bot-platform/>

Overcoming Software Supply Chain Attacks

<https://blog.karambit.ai/overcoming-software-supply-chain-attacks-c8746a0236ab>

iconburst NPM supply chain attack

<https://www.scmagazine.com/news/iconburst-supply-chain-attack-uses-typo-squatting-to-spread-malicious-javascript-packages-via-npm>

Deceptive Deprecation: The Truth About npm Deprecated Packages

<https://blog.aquasec.com/deceptive-deprecation-the-truth-about-npm-deprecated-packages>

aquasec/CIS supply chain security guide

<https://www.aquasec.com/news/software-supply-chain-security-guide-cis-aqua-security/>

OWASP kube top ten risks #2: supply chain vulnerabilities

<https://github.com/OWASP/www-project-kubernetes-top-ten/blob/main/2022/en/src/K02-supply-chain-vulnerabilities.md>

Git Checkout Authentication to the Rescue of Supply Chain Security

[https://archive.fosdem.org/2023/schedule/event/security\\_where\\_does\\_that\\_code\\_come\\_from/](https://archive.fosdem.org/2023/schedule/event/security_where_does_that_code_come_from/)

Software supply chain security practices are maturing – but it's a work in progress

<https://www.reversinglabs.com/blog/openssf-survey-supply-chain-security-practices>

Open Source Supply Chain Security at Google

<https://research.swtch.com/acmscored>

CVE Half-Day Watcher

<https://github.com/Aqua-Nautilus/CVE-Half-Day-Watcher>

Few Open Source Projects are Actively Maintained

<https://www.infoworld.com/article/3708630/report-finds-few-open-source-projects-actively-maintained.html>

The Massive Bug at the Heart of NPM

<https://blog.vlt.sh/blog/the-massive-hole-in-the-npm-ecosystem>

A Study on Navigating Open-Source Dependency Abandonment:

<https://courtney-e-miller.github.io/static/media/WeFeelLikeWereWingingIt.dc3c76d3b3c2d12f4fe.pdf>

# SBOM Reading List

Making Better SBOMs

<https://kccnca2022.sched.com/event/182GT/>

<https://www.youtube.com/watch?v=earq775L4fc>

Reflections on Trusting Trust

[https://www.cs.cmu.edu/~rdriley/487/papers/Thompson\\_1984\\_ReflectionsonTrustingTrust.pdf](https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf)

<https://web.mit.edu/6.033/2002/wwwdocs/handouts/h25-review2slides2.pdf>

Introduction to SBOMs - What is it and do I need one?

<https://www.youtube.com/watch?v=jVI6K5h6PzY>

Generate sboms with synt and jenkins

[https://www.youtube.com/watch?v=nMLveJ\\_TxA](https://www.youtube.com/watch?v=nMLveJ_TxA)

Profound Podcast - Episode 10 (John Willis and Josh Corman)

<https://www.buzzsprout.com/1758599/8761108-profound-dr-deming-episode-10-josh-corman-captain-america>

GitHub Self-Service SBOMs

<https://github.blog/2023-03-28-introducing-self-service-sboms/>

Do SBOMS Need VEX?:

[https://www.linkedin.com/posts/aph10\\_sbom-software-supply-chain-security-vex-activity-7108017924384137216-VARV/](https://www.linkedin.com/posts/aph10_sbom-software-supply-chain-security-vex-activity-7108017924384137216-VARV/)



# Glossary

- CVE - Common Vulnerabilities and Exposures - <https://cve.mitre.org/>
- CVSS - Common Vulnerability Scoring System - <https://nvd.nist.gov/vuln-metrics/cvss>
- CISA - cybersecurity and infrastructure security agency - <https://cisa.gov>
- KEV - Known Exploited Vulnerabilities <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- EPSS - Exploit Prediction Scoring System - <https://www.first.org/epss/>
- SBOM - Software Bill of Materials - <https://www.cisa.gov/sbom>
- VEX - Vulnerability Exploitability eXchange - <https://github.com/openvex/spec>
- CSAF - Common Security Advisory Framework - <https://oasis-open.github.io/csaf-documentation/>
- GHSA - GitHub Security Advisory - <https://github.com/advisories>
- OpenSSF - Open Source Security Foundation - <https://openssf.org/>

