

The Lessons of Log4shell

Preparing for the Next Zero-Day

Hello World



Paul Novarese
Principal Solutions Architect
Anchore, Inc.

pvn@anchore.com

Fediverse: @pvn@mas.to

Agenda

00

The Silver Bullet

01

Lesson 1

02

Lesson 2

03

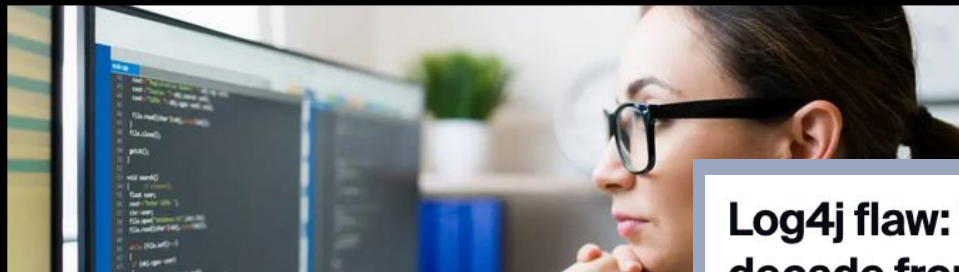
Lesson 3

This is not a utopian talk with magic solutions



Lesson 1: The Bad News

With 40% of Log4j Downloads Still Vulnerable, Security Retrofitting Needs to Be a Full-Time Job



Log4j flaw: Why it will still be causing problems a decade from now

Log4Shell ain't over until it's over, warns the US review board tasked with investigating the critical Apache Log4J flaw known as Log4Shell.



Written by **Liam Tung**, Contributing Writer on July 15, 2022

WE ARE NEVER EVER EVER

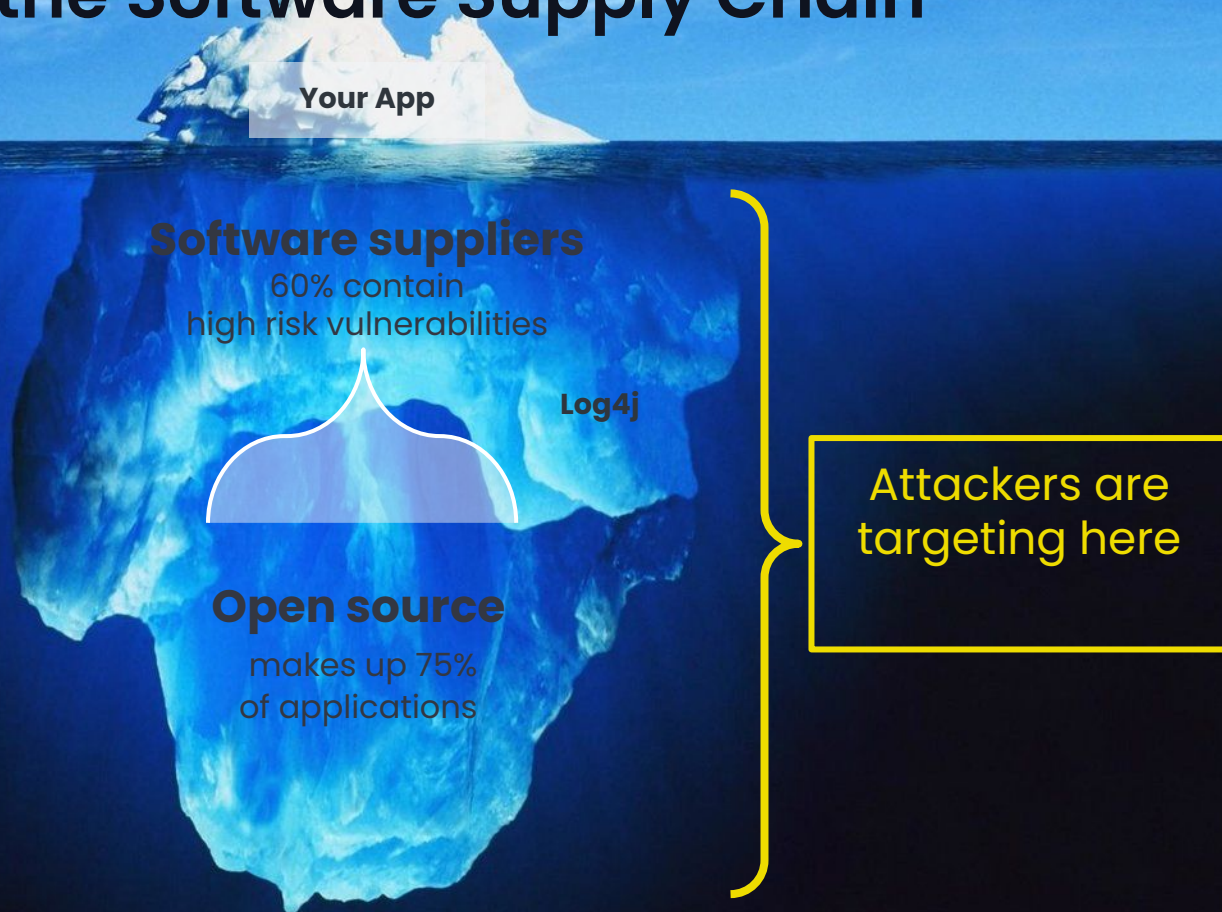
GETTING RID OF LOG4SHELL

Lesson 2:

What We Don't Know

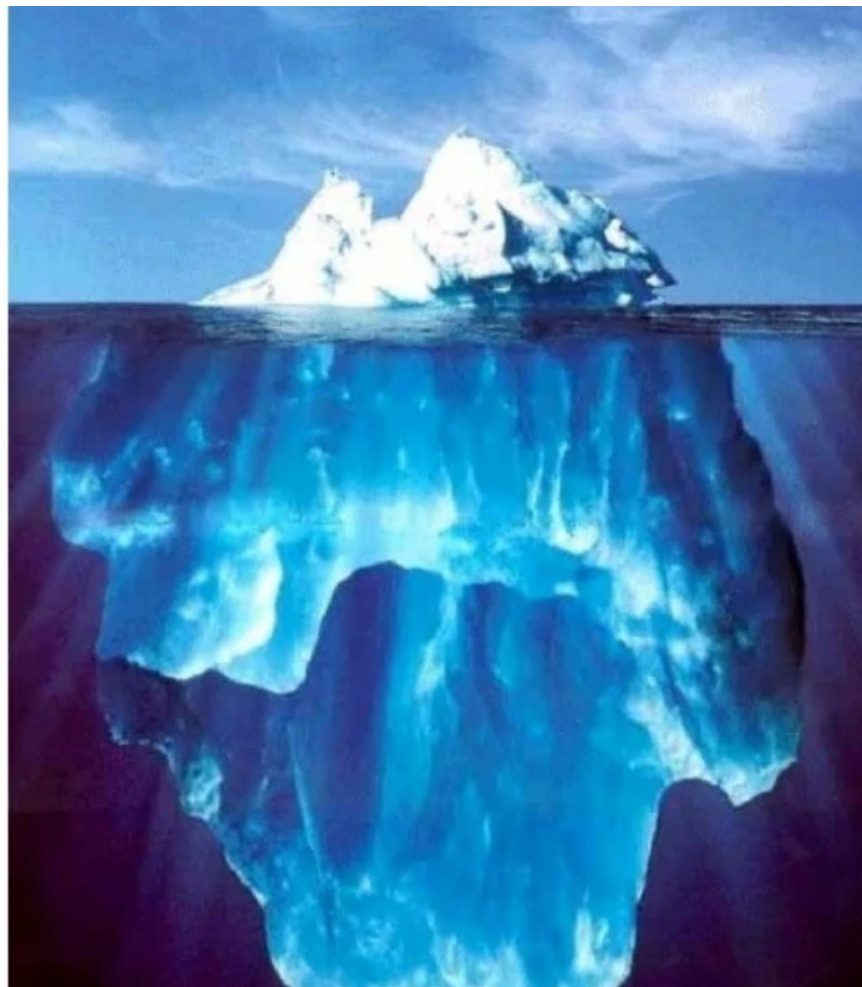
Hidden Risk in the Software Supply Chain

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
METRO Webinar
October 6, 2009



An iceberg floating in a blue ocean under a blue sky with light clouds. The tip of the iceberg is above the water line, while the much larger base is submerged. A white semi-circular outline is drawn on the submerged part of the iceberg. A yellow bracket on the right side of the iceberg spans from the tip down to the submerged section.

**Direct
Dependencies**

**Transitive
Dependencies**

Attackers are
targeting here

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

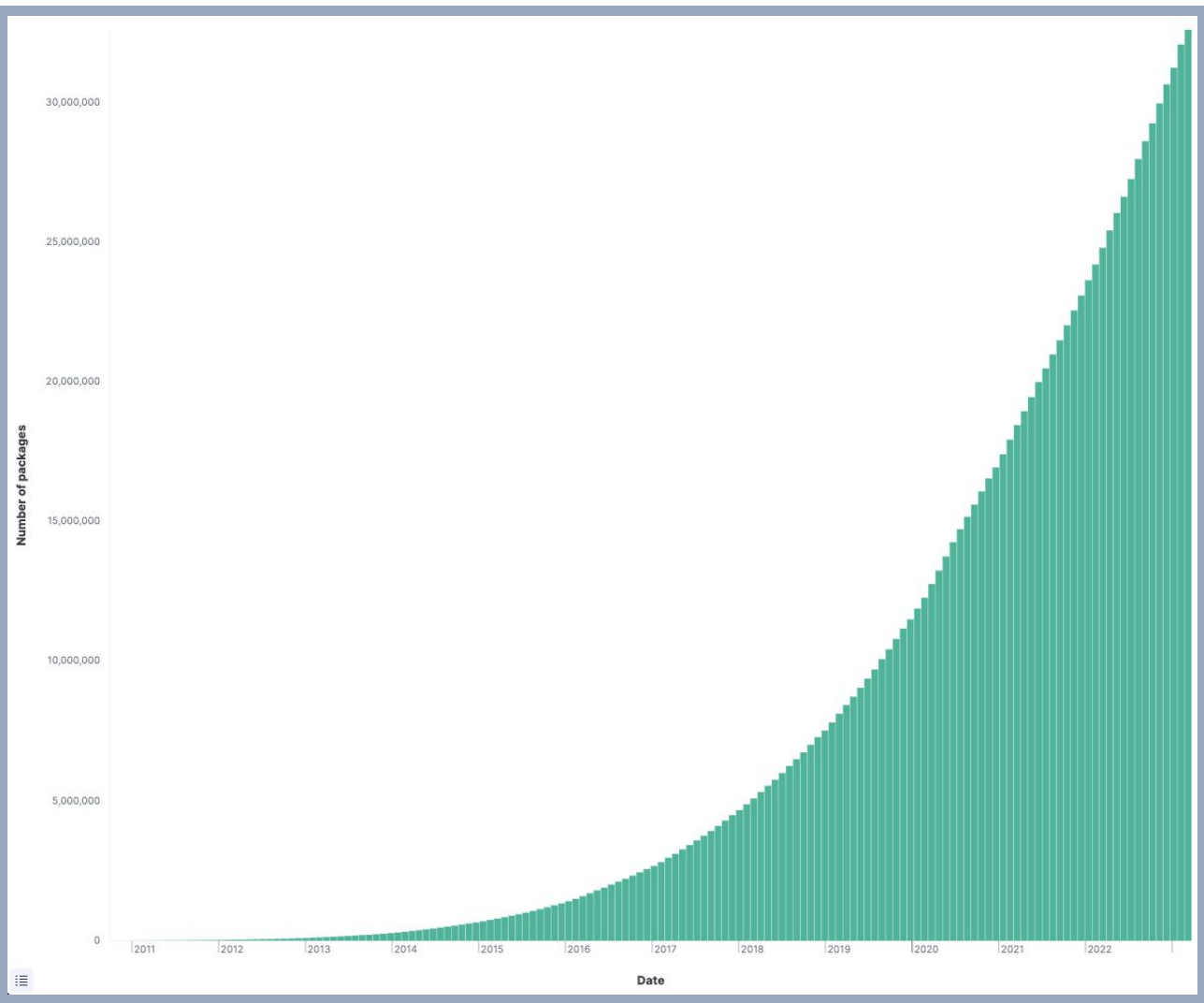
WHY Don't We Know?

- Explosion of Open Source
- Transitive Dependencies

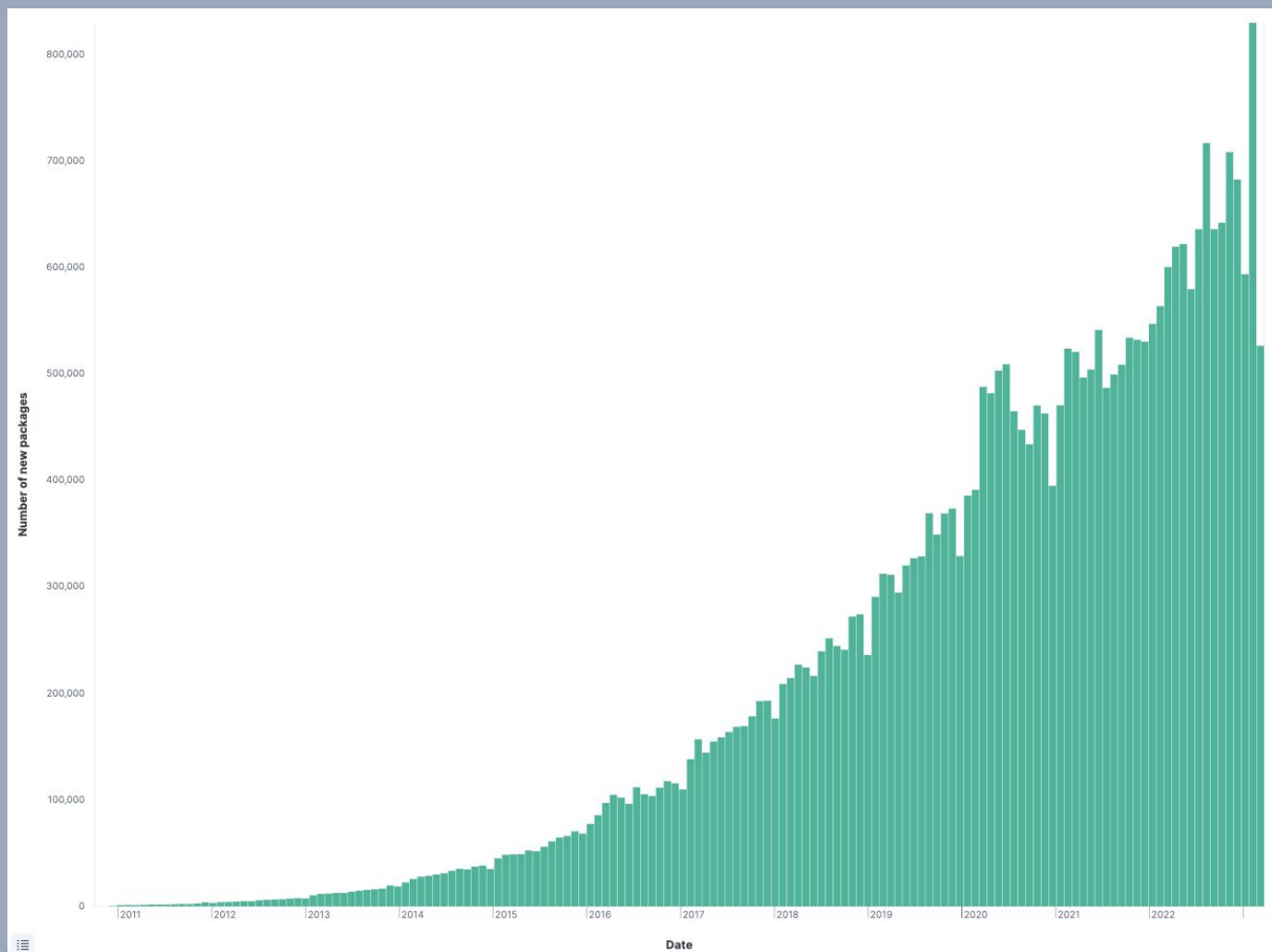
The Result:



Number of NPM packages



Number of NEW packages

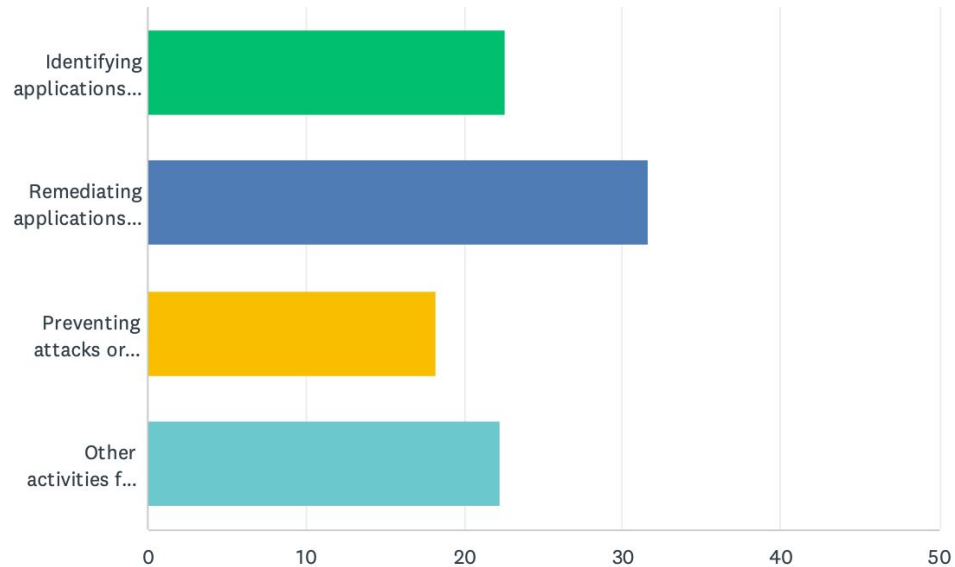


Open source is huge

- NPM introduced 2010
- 32 million packages (as of March 2023)
- Approx. 1,000,000 new packages **per month**
- That's just NPM!

Q12 Estimate how many hours you personally have spent to date on each of the following activities.

Answered: 195 Skipped: 15



If We Knew...

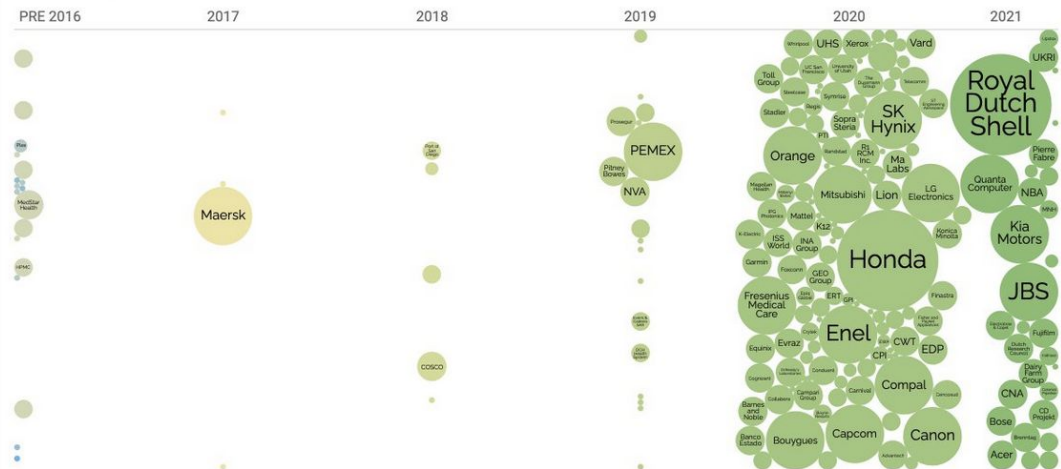
- People spent insane amounts of time just finding log4j, because nobody knew where (or even if) it was hiding
- Knowing = Faster Remediation
- SBOMs help, a LOT, but...
 - They aren't a silver bullet
 - Scanners aren't perfect (e.g. can't penetrate binary blobs, cf. OpenSSL3.)
 - Not all SBOMs are equal
 - 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>
 - I don't know what the end game is but generating them is better than nothing, figure out the details later



interactive: bit.ly/3h1IYPs



size = size of organisation



David McCandless, Swarna Maslekar
Information is Beautiful

sources: bleeping computer, zdnet, forbes, BBC
& other news reports // 23rd June 2021

The predictable consequence

- Ransomware has exploded along with transitive dependencies and open source in general
- I don't believe in coincidences

Lesson 3:

The Other Thing

We Don't Know



Paul Novarese (He/Him) · You

Software Supply Chain Security at Anchore

1yr · Edited ·



The [#log4j](#) debacle is going to have ramifications far beyond the vulnerability itself. There has been a lot of inertia in how issues are evaluated and classified, how information about those issues is disseminated, and how organizations respond to them, and [#log4shell](#) has exposed a lot of these problems. This will be a catalyst for a lot of changes that are way overdue.



April King

@CubicleApril



The fact that there are almost 10,000 CVEs with the same CVSS score as the Log4j vulnerability suggests to me that maybe the scale should be logarithmic.

6:26 PM · Dec 11, 2021 · Twitter for iPhone

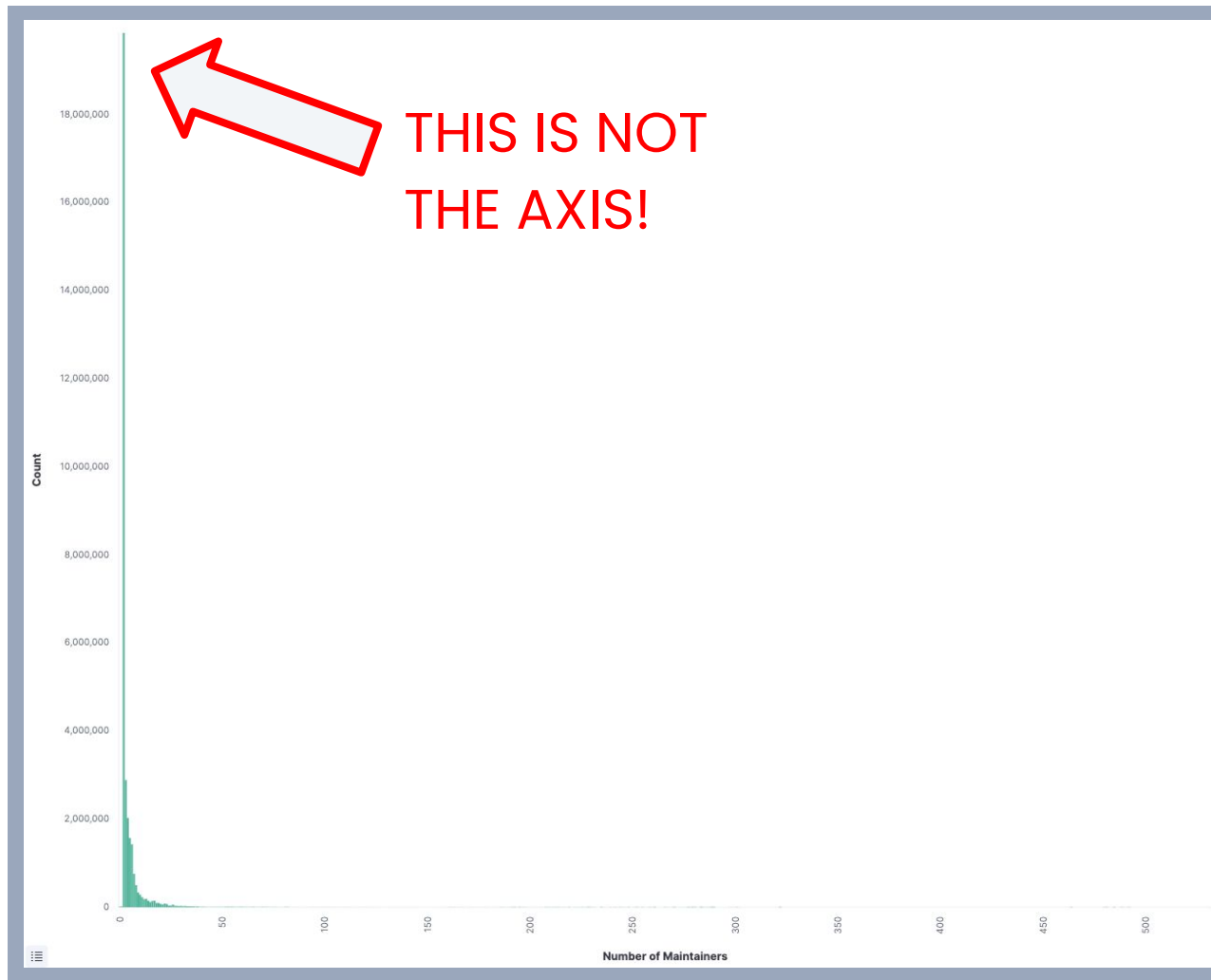
71 Retweets **6** Quote Tweets **736** Likes

OK, If Not CVSS, Then What?

- GHSA's (more transparent than CVEs)
- CISA KEV
- EPSS
- GitHub Insights and other project health metrics
 - This is (currently) a very manual process
 - But it's getting a lot easier
 - Project health isn't **directly** about safety
 - What happens when it hits the fan?

An Example Project
Health Metric:

Number of
Maintainers



Recap, Notes, &c.

Recap

- Log4Shell is Radioactive, Immortal
- We Don't Know What's In Our Own Software
- We Must Get Better Guidance
- How Software Gets Made is Different Now
- We Have a Lot of Eggs in GitHub's Basket
- Think about Risk in the General Case

Footnotes

Slide 6 (Log4Shell will still be causing problems a decade from now):

<https://www.zdnet.com/article/log4j-flaw-why-it-will-still-be-causing-problems-a-decade-from-now/>

Slide 6 (40% of Log4j downloads still vulnerable):

<https://securityintelligence.com/articles/log4j-downloads-vulnerable/>

Slide 10 (possible origin of the iceberg):

<https://www.slideshare.net/loriayre/open-source-library-system-software-free-is-just-the-tip-of-the-iceberg>

Slides 12,13, 22 (Open Source is Bigger Than You Can Imagine):

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

Slide 14 (log4j survey etc):

<https://anchore.com/log4j/>

Slide 18:

<https://twitter.com/CubicleApril/status/1469825942684160004>

https://www.linkedin.com/posts/novarese_log4j-log4shell-activity-6876206319238463488-8bEA

Reading List

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>

SBOM Everywhere:

<https://github.com/ossf/sbom-everywhere>

My previous DevOpsDays 2022 talk (Learn From Log4Shell):

https://www.youtube.com/watch?v=PINtIL_oN0k
<https://github.com/pvnovarese/2022-devopsdays>

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/>

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/>

CISA Known Exploited Vulnerability Catalog

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

Exploit Prediction Scoring System

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

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>

SBOM Takeaways

00

SBOMs enable continuous, automated security/compliance checks, reduce time spent identifying and remediating issues

01

SBOMs improve a lot of things but do not solve every problem you have

02

Log4j is extremely easy to find, OpenSSL 3 is often obscured

03

SBOMs are more effective when created by maintainers rather than consumers, but something is better than nothing

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

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>

Q&A

Download Syft

<https://github.com/anchore/syft>

Download Grype

<https://github.com/anchore/grype>

Let us know if you like it by giving us a star on GitHub

Get an invite to our open source community Slack at
<https://anchore.com/slack/>

These slides and lab examples archived here:

<https://github.com/pvnovarese/2023-09-lessons-of-log4shell>