

# The Lessons of Log4shell

Preparing for the Next Zero-Day

# Hello World



Paul Novarese  
Principal Solutions Architect  
Anchore, Inc.

[pvn@anchore.com](mailto:pvn@anchore.com)

Fediverse: @pvn@mas.to

**This is not a utopian talk with magic solutions**



# Agenda

01

When

02

What

03

Who

04

How

# Lesson 1: When



**Fukushima Daiichi  
Incident: 2011  
Cleanup: at LEAST thirty years**







**Chernobyl  
Incident: 1986  
Cleanup: at LEAST until 2065**

# With 40% of Log4j Downloads Still Vulnerable, Security Retrofitting Needs to Be 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 Tynes, Contributing Writer on July 15, 2022



**Mark Chmarny** (He/Him) • Following

Product, Infra & DevEx at Cruise

4mo • 🌐



29% of Log4j consumption worldwide STILL uses versions that are known to be vulnerable (source: [Sonatype](#))



**WE ARE NEVER EVER EVER**

**GETTING RID OF LOG4SHELL**

# Lesson 2: What

# Hidden Risk in the Software Supply Chain

Your App

**Software suppliers**

60% contain  
high risk vulnerabilities

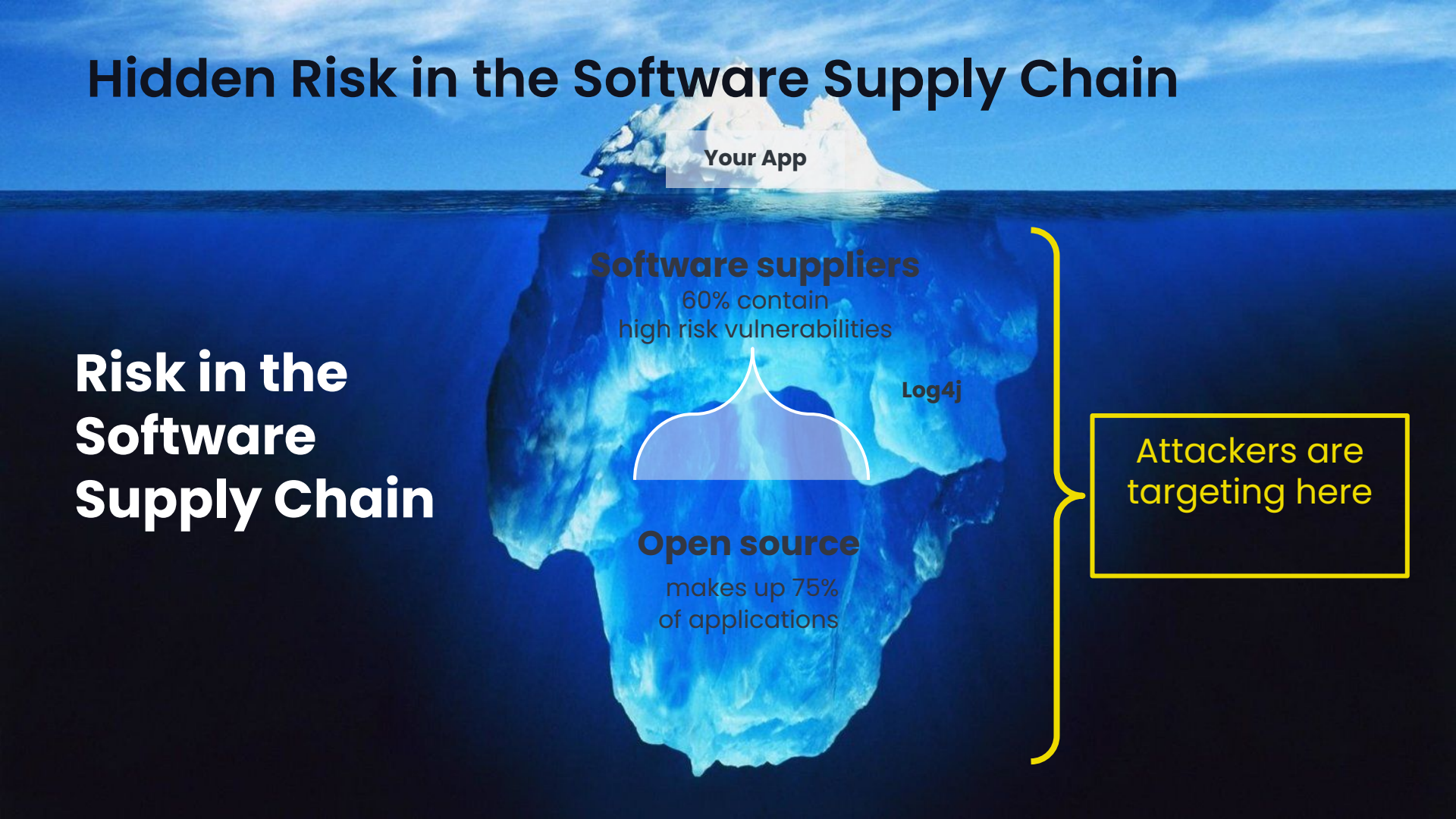
Log4j

**Open source**

makes up 75%  
of applications

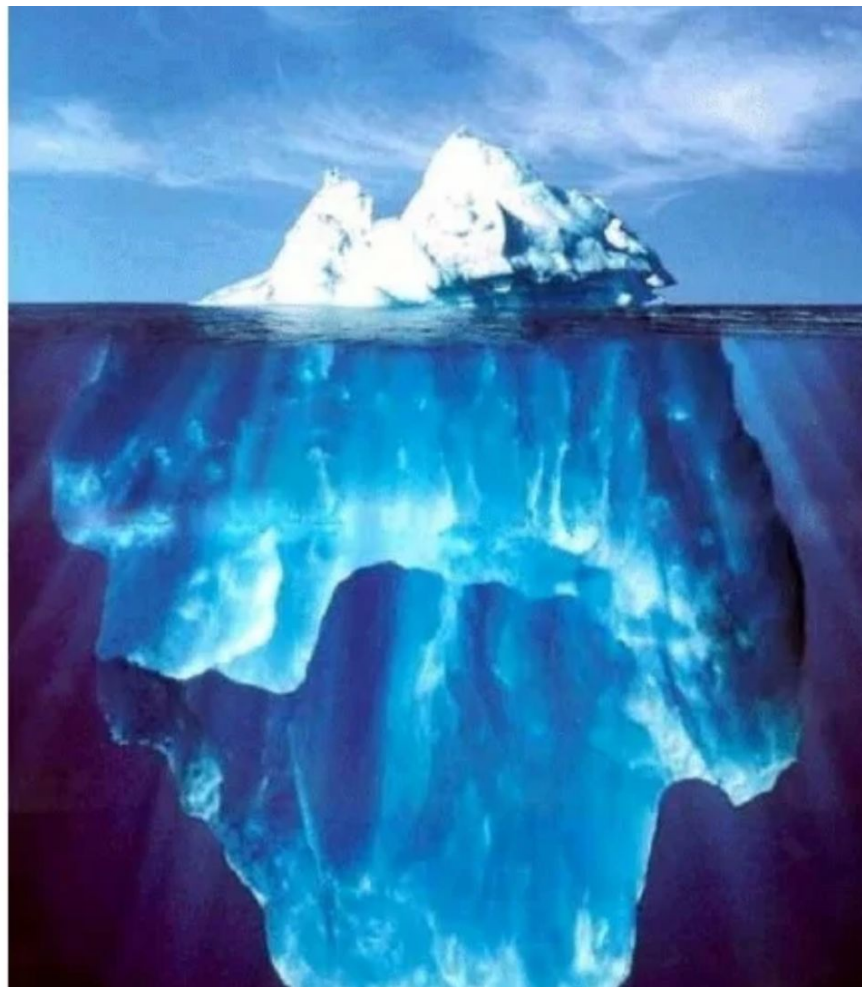
Attackers are  
targeting here

**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 the ocean. The tip of the iceberg is above the water line, and the much larger base is submerged. A white semi-circular shape 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 part. The sky is blue with some clouds, and the water is a deep blue.

**Direct  
Dependencies**

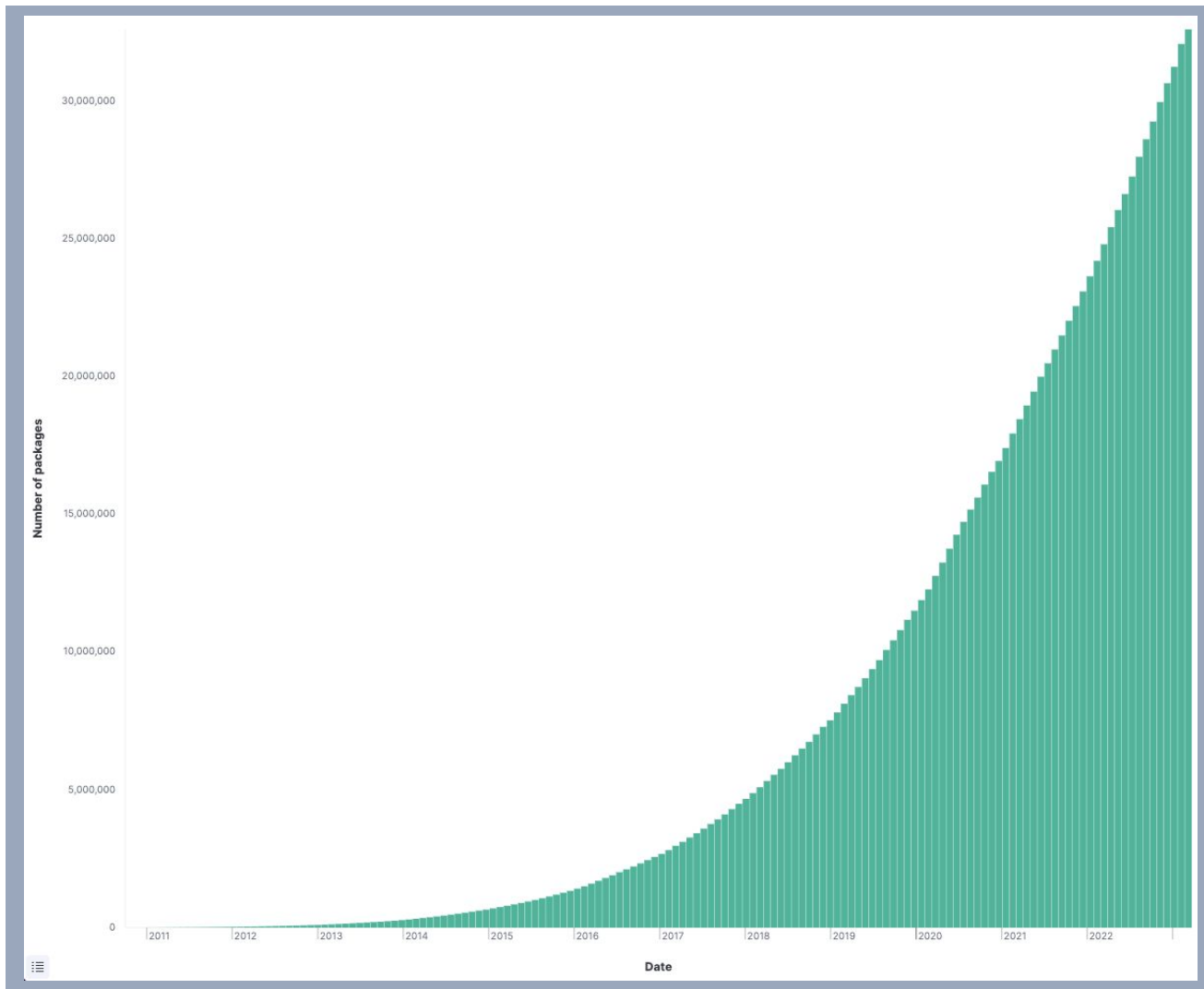
**Transitive  
Dependencies**

Attackers are  
targeting here

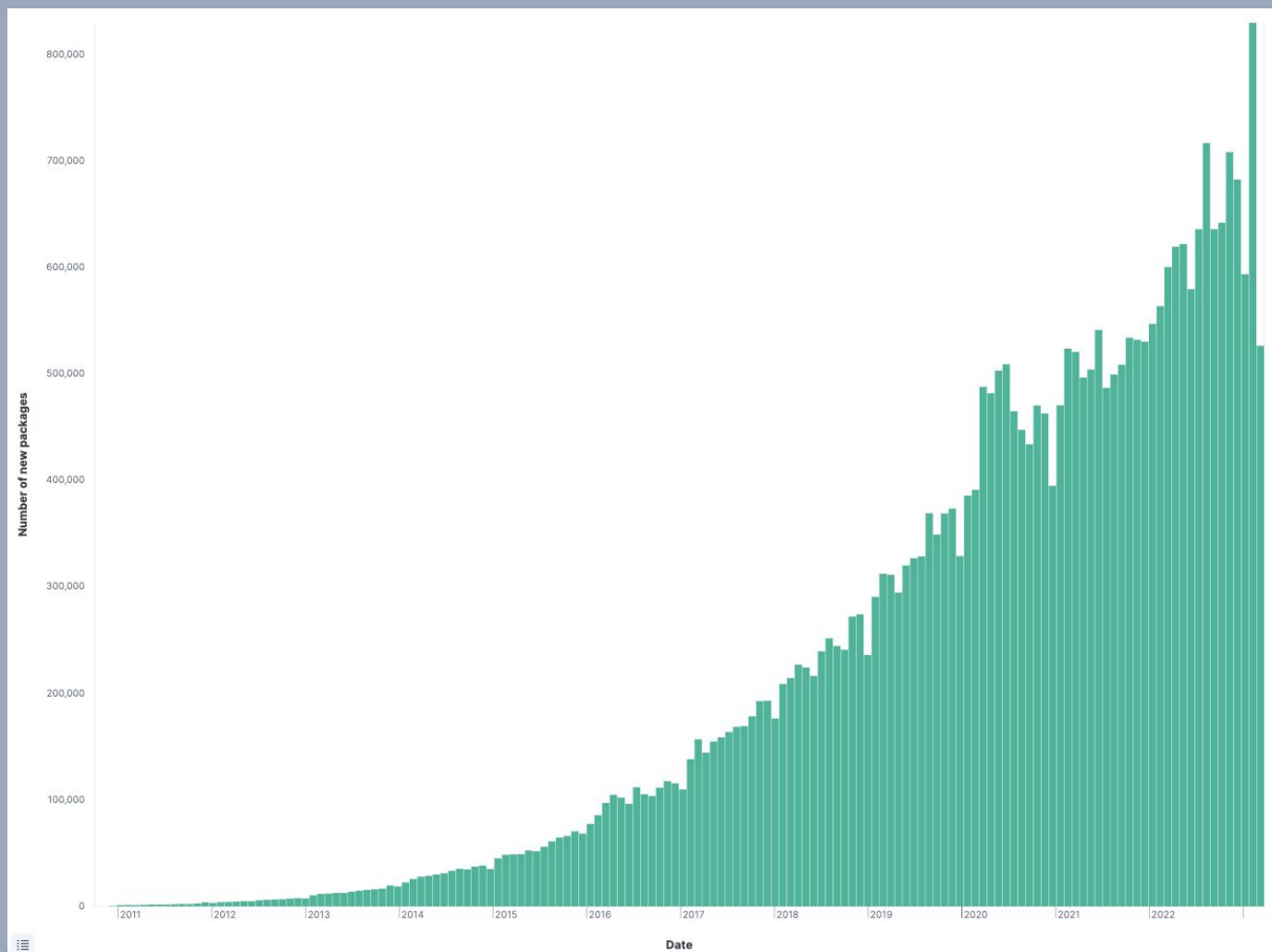
# This metaphor...

- 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

# Number of NPM packages



# Number of NEW packages





# Open source is huge

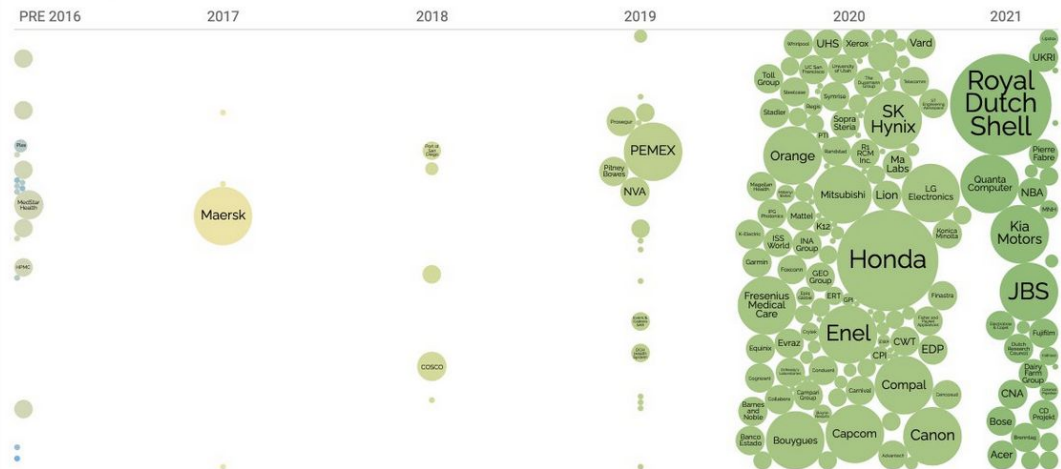
- NPM introduced 2010
- 32 million packages (as of March 2023)
- Approx. 1,000,000 new packages \*\*per month\*\*
- That's just NPM!



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



size = size of organisation



David McCordless, Swanuja Maslekar  
**Information is Beautiful**

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

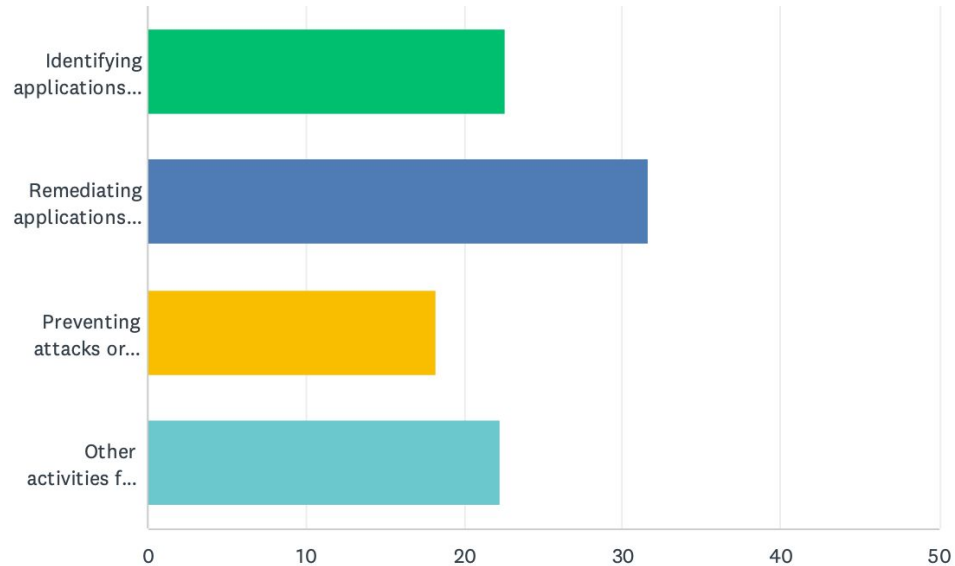
# The predictable consequence

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



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

Answered: 195 Skipped: 15





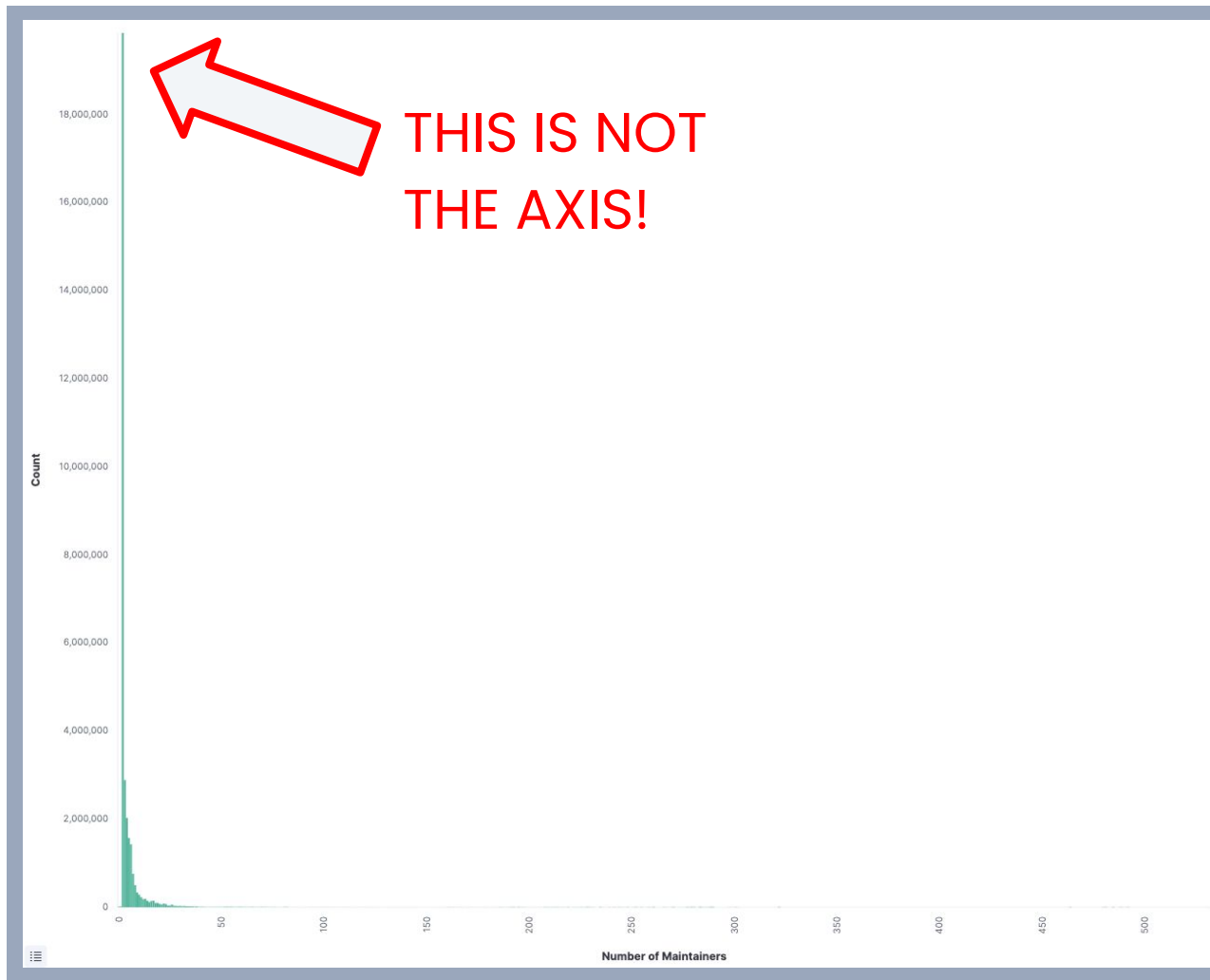
# If We Knew What We are Consuming

- 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...
  - 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

# Lesson 3: Who

An Example Project  
Health Metric:

Number of  
Maintainers





**daniel:// stenberg://**

@bagder



If you are a multi billion dollar company and are concerned about log4j, why not just email OSS authors you never paid anything and demand a response for free within 24 hours with lots of info? (company name redacted for \*my\* peace of mind)

Dear Haxx Team Partner,

You are receiving this message because [REDACTED] uses a product you developed. We request you review and respond within 24 hours of receiving this email. If you are not the right person, please forward this message to the appropriate contact.

As you may already be aware, a newly discovered zero-day vulnerability is currently impacting Java logging library Apache Log4j globally, potentially allowing attackers to gain full control of affected servers.

The security and protection of our customers' confidential information is our top priority. As a key partner in serving our customers, we need to understand your risk and mitigation plans for this vulnerability.

Please respond to the following questions using the template provided below.



# Stop thinking about open source like a vendor

This



Not this



# Lesson 4:

## How



**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

## CVE-2020-19909

On August 25 2023, we got [an email to the curl-library mailing list](#) from Samuel Henrique that informed us that “someone” had recently created a CVE, a security vulnerability identification number and report really, for a curl problem.

```
I wanted to let you know that there's a recent
curl CVE published and it doesn't look like it
was acknowledged by the curl authors since it's
not mentioned in the curl website: CVE-2020-19909
```

We can't tell who filed it. We just know that it is now there.



anchore / syft

Q Type to search



<> Code Issues **251** Pull requests **17** Actions Projects Security **1** Insights

Pulse

Contributors

Community Standards

Commits

Code frequency

Dependency graph

Network

Forks

September 4, 2023 – September 11, 2023

Period: 1 week ▾

### Overview

20 Active pull requests

13 Active issues

16

Merged pull requests

4

Open pull requests

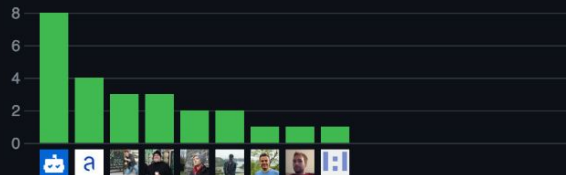
6

Closed issues

7

New issues

Excluding merges, **9 authors** have pushed **16 commits** to main and **21 commits** to all branches. On main, **20 files** have changed and there have been **240 additions** and **124 deletions**.



1 Release published by 1 person

**v0.90.0**

published 3 hours ago

16 Pull requests merged by 7 people

**fix the help output of power-user**

#2113 merged 8 hours ago

# OK, If Not CVSS, Then What?

- GHSA's (more transparent than CVEs)
- CISA KEV, EPSS, VEX &c
- 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?



# Recap, Notes, &c.

# Recap

- Log4Shell is radioactive and immortal
- How software gets made has changed
- We don't know what's in our software
- We don't know who is supplying it
- We have to change how we evaluate it
- GitHub is uniquely positioned
- Think about risk in the general case

# Footnotes

Log4Shell's immortality:

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

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

possible origin of the iceberg:

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

Open Source is Bigger Than You Can Imagine::

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

log4j survey etc:

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

Various tweets &c:

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

[https://www.linkedin.com/posts/novarese\\_log4j-log4shell-activity-6876206319238463488-8bEA](https://www.linkedin.com/posts/novarese_log4j-log4shell-activity-6876206319238463488-8bEA)

<https://twitter.com/bagder/status/1484672924036616195>

<https://lists.haxx.se/pipermail/daniel/2023-September/000032.html>

# 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>

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/>

I am not a Supplier:

<https://www.softwaremaxims.com/blog/not-a-supplier>

<https://opensourcesecuritypodcast.libsyn.com/episode-365-i-am-not-your-supplier-with-thomas-depierre>

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/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/>

CISA Known Exploited Vulnerability Catalog

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

Exploit Prediction Scoring System

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

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/)

# 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](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](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/>

# 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>