The Legacy of

(And the Future of DevSecOps)

Texas Linux Fest
2024-04-13

This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International
Paul Novarese, Solutions Engineer, Anchore

Paul Novarese
Principal Solutions Architect
Anchore, Inc.
pvn@anchore.com
Fediverse: @pvn@mas.to
Paul Novarese, Solutions Engineer, Anchore
This is not a utopian talk with magic solutions
Agenda

01 Log4Shell Today
02 A Fundamental Shift
03 Software “Supply” Chains
04 Three Big Things
Learn From Log4Shell
I posted this in December 2021 right after log4shell dropped.

Almost a year later, October 2022 I presented “learn from log4shell” at devopsdays houston, I had basically given up and said this was completely wrong.

Now it may end up being correct, but not in the way I thought.

https://twitter.com/CubicleApril/status/1469825942684160004
https://www.linkedin.com/posts/novarese_log4j-log4shell-activity-6876206319238463488-8bEA

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.

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.
The State of Log4Shell Today
It's been over two years and log4shell is still the single most exploited CVE.

Data from CISA - cybersecurity and infrastructure security agency https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-279a

### Table I: Top CVEs most used by Chinese state-sponsored cyber actors since 2020

<table>
<thead>
<tr>
<th>Vendor</th>
<th>CVE</th>
<th>Vulnerability Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Log4j</td>
<td>CVE-2021-44228</td>
<td>Remote Code Execution</td>
</tr>
<tr>
<td>Pulse Connect Secure</td>
<td>CVE-2019-11510</td>
<td>Arbitrary File Read</td>
</tr>
<tr>
<td>GitLab CE/EE</td>
<td>CVE-2021-22205</td>
<td>Remote Code Execution</td>
</tr>
<tr>
<td>Atlassian</td>
<td>CVE-2022-26134</td>
<td>Remote Code Execution</td>
</tr>
<tr>
<td>Microsoft Exchange</td>
<td>CVE-2021-26855</td>
<td>Remote Code Execution</td>
</tr>
</tbody>
</table>
Fukushima Daiichi Incident: 2011
Cleanup: at LEAST thirty years
Chernobyl Incident: 1986
Cleanup: at LEAST until 2065
In 2023: 40% of Log4j downloads still vulnerable. According to some experts, it will still be causing problems a decade from now.

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.

Mark Chmarny (He/Him) · Following
Product, Infra & DevEx at Cruise
4mo · $5

87% of Log4j consumption worldwide STILL uses versions that are known to be vulnerable (source: Sonatype)
Log4Shell Highlighted a Fundamental Shift
Hidden Risk in the Software Supply Chain

Risk in the Software Supply Chain

- Software suppliers: 60% contain high risk vulnerabilities
- Open source: makes up 75% of applications
- Log4j

Attackers are targeting here
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
Attackers are targeting here

Direct Dependencies

Transitive Dependencies

Attackers are targeting here
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
Number of NPM packages
Number of NEW packages
How big is big? (all packages over time)

8.1 million
How big is big? (all versions of all packages)

93 million
Also 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
3,732,919 packages
42,958,444 versions
850,084 maintainers
231,488 namespaces
752,313 keywords
256,314,168,001 downloads
Are #Ransomware attacks increasing? I think #Ransomware attacks are increasing...

interactive: bit.ly/3h1lYPs
The predictable consequence

- Ransomware has exploded along with transitive dependencies and open source in general
- I don’t believe in coincidences
Attackers are targeting here
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
- I don’t know what the end game is but generating them is better than nothing, figure out the details later
Software “Supply” Chains
Software “Supply” Chain: The Iceberg Funnel
The Reverse Funnel

Compromised Package

Open Source

Public Image

Base Image

Docker Image

Commercial Software

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source

Open Source
What is an SBOM?
An Example Project
Health Metric:

Number of Maintainers

THIS IS NOT THE AXIS!
How many people are maintaining these things?
100,000 downloads
> 1,000,000 downloads
How many packages are more than a year old?
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 [Company Name] 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.
**log4shell Timeline**

- **24 Nov**: Initial disclosure to ASF by Chen Zhaojun (Alibaba)
- **29 Nov**: log4j PR #608
- **1 Dec**: First evidence of exploit in the wild (per Cloudflare)
- **6 Dec**: log4j 2.15.0-rc1 ships
- **9 Dec**: Discussion of exploit on minecraft forums, LunaSec coins "Log4Shell", &c
- **10 Dec**: Official public disclosure CVE-2021-44228
- **13 Dec**: Log4j 2.16.0
- **17 Dec**: Log4j 2.17.1
Stop thinking about open source like a vendor

This

Not this
60% of maintainers describe themselves as unpaid hobbyists

Which of the following phrases best describes how you approach your role as an open source maintainer?

- 23% I’m a semi-professional maintainer, and earn some of my income from maintaining projects
- 13% I’m a professional maintainer, and earn most of my income from maintaining projects
- 14% I’m an unpaid hobbyist and do not want to get paid for maintaining projects
- 46% I’m an unpaid hobbyist, but would appreciate getting paid for maintaining projects

n=326

Credit: Tidelift 2023 Open Source Maintainer Survey
Summary of Software Supply Chains

- Red Hat is a supplier - 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.
Breaking News
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.

BLEEPINGCOMPUTER

Top.gg supply chain attack highlights subtle risks

Threat actors used fake Python infrastructure and cookie stealing to poison multiple GitHub code repositories, putting another spotlight on supply chain risks.

Hackers poison source code from largest Discord bot platform

By Bill Toulas
March 2024 was wild

- NVD Chaos (started mid-Feb, noticed early March)
- top.gg python-sdk poisoned (discovered mid March)
- xz backdoor (discovered late March)
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.
Top.gg supply chain attack highlights subtle risks

Threat actors used fake Python infrastructure and cookie stealing to poison multiple GitHub code repositories, putting another spotlight on supply chain risks.

Hackers poison source code from largest Discord bot platform

By Bill Toulas
JIA TAN
Three Big Things
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.
CVE by year
Big Thing #1: If Not CVE/CVSS, Then What?

- GHSAs (more transparent than CVEs)
- CISA KEV, EPSS, VEX, CSAF, &c
- OpenSSF Malicious Packages Repository
- GitHub Insights and other project health metrics
  - This is (currently) a very manual process
  - But it’s getting a lot easier
Big Thing #2: Project Health/Insights

- This is PROACTIVE (the better advisory data, scoring etc is about reactive improvements)
- This is (currently) a manual process (getting easier)
- Evaluating project health isn’t directly about safety, it’s about tracking all of those deps in the iceberg,
- Are the projects you’re depending on healthy, will you be able to work with them?
September 4, 2023 – September 11, 2023

Overview

Pulse

Contributors
Community Standards
Commits
Code frequency
Dependency graph
Network
Forks

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
Open Source has gotten so big that opportunistic, financially-motivated attackers are extremely incentivized to focus on it.

Supply chain attacks are reusable.

Even state-sponsored attackers can’t ignore it.

Scale means that it’s often useful even to get to particular targets.
Twitter was incredibly central to Log4Shell reaction, forming consensus, and generally just figuring out what was happening.

If log4shell dropped today, this reaction/recovery would be notably worse because of infosec splintering to (e.g.) mastodon, linkedin, bsky, threads.

None of these networks have the critical mass that Twitter had and it doesn’t seem to be improving.
This seems really bad

1. Well it’s not great
2. But things are mostly working OK
3. Open source adapts
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
Big Changes

1. Better metrics and data sources are coming
2. Tracking dependencies is more proactive
3. Supply chain attacks are here to stay
4. Twitter is Over
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
The (first two) “big things” are still very embryonic and probably not ready for prime time but tools are starting to adopt a lot of this

SBOMs: https://github.com/anchore/syft
Vulnerabilities: https://github.com/anchore/grype
Webinars: https://anchore.com/webinars/
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
- Try to be proactive
Q&A

Our open source projects:

https://github.com/anchore/syft
https://github.com/anchore/grype
https://github.com/anchore/grant

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

These slides are archived:
https://github.com/pvnovarese/2024-04-legacy-of-log4shell
Notes, &c.
Footnotes

Package data - [https://ecosyste.ms/](https://ecosyste.ms/)
Open Source is Bigger Than You Can Imagine - [https://anchore.com/blog/open-source-is-bigger-than-you-imagine/](https://anchore.com/blog/open-source-is-bigger-than-you-imagine/)
log4j survey etc - [https://anchore.com/log4j/](https://anchore.com/log4j/)
possible origin of the iceberg - [https://www.slideshare.net/loriayre/open-source-library-system-software-free-is-just-the-tip-of-the-iceberg](https://www.slideshare.net/loriayre/open-source-library-system-software-free-is-just-the-tip-of-the-iceberg)
xz logo: [https://infosec.exchange/@jerry/t12186387514069376](https://infosec.exchange/@jerry/t12186387514069376)

Log4Shell's immortality:
[https://securityintelligence.com/articles/log4j-downloads-vulnerable/](https://securityintelligence.com/articles/log4j-downloads-vulnerable/)

Patrick Garrity discussing EPSS and Improved Metrics:

Various tweets &c:
[https://twitter.com/CubicleApril/status/1469825942684160004](https://twitter.com/CubicleApril/status/1469825942684160004)
[https://twitter.com/bagder/status/1484672924036616195](https://twitter.com/bagder/status/1484672924036616195)
[https://lists.haxx.se/pipermail/daniel/2023-September/000032.html](https://lists.haxx.se/pipermail/daniel/2023-September/000032.html)
Projects and Data Sources

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


Open Source Insights: https://deps.dev/
Reading List

Filling the NVD data gap
https://github.com/anchore/nvd-data-overrides

NVD Chaos Podcast
https://resilientcyber.substack.com/p/s8e11-josh-bressers-and-dan-lorenz

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:

I am not a Supplier:
https://www.softwaremaxims.com/blog/not-a-supplier

My previous DevOpsDays 2022 talk (Learn From Log4Shell):
https://www.youtube.com/watch?v=PINIL_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/

Do SBOMs Need VEX?:

A Study on Navigating Open-Source Dependency Abandonment:
https://courtney-e-miller.github.io/static/media/WeFeelLikeWereWingingIt.dc3c76d3b3c2d1214fee.pdf

Shedding Light on CVSS Scoring Inconsistencies:
https://arxiv.org/abs/2308.15259
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/AgPahhBr9xHMAPWi

OSS backdoors: the folly of the easy fix

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://robmenschning.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/111348817f163534252

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 jai tan’s commit timestamps
https://twitter.com/birchb9y/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

Lessons from XZ Utils: Achieving a More Sustainable Open Source Ecosystem
Supply Chains Reading List

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

Deceptive Deprecation: The Truth About npm Deprecated Packages

aquasec/CIS supply chain security guide

OWASP kube top ten risks #2: supply chain vulnerabilities

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

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

State of the Software Supply Chain:

Few Open Source Projects are Actively Maintained:

The Massive Bug at the Heart of NPM:
https://blog.vlt.sh/blog/the-massive-hole-in-the-npm-ecosystem
Dealing with log4shell (detection, mitigation, workarounds):

Keeping up with log4shell (post mortem)

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
Glossary

CVE - Common Vulnerabilities and Exposures - https://cve.mitre.org/
CISA - cybersecurity and infrastructure security agency - https://cisa.gov
VEX - Vulnerability Exploitability eXchange - https://github.com/openvex/spec
GHSA - GitHub Security Advisory - https://github.com/advisories
OpenSSF - Open Source Security Foundation - https://openssf.org/
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


Introduction to SBOMs – What is it and do I need one? - https://www.youtube.com/watch?v=jVI6K5h6PzY

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

Profound Podcast – Episode 10 (John Willis and Josh Corman):

GitHub Self-Service SBOMs: https://github.blog/2023-03-28-introducing-self-service-sboms/