anchore

Learn from Log4shell

Using SBOMs for Zero-Day Preparadness

Hello World

Paul Novarese

Principal Solutions Architect

Anchore, Inc.

pvn@anchore.com - @pvn

@pvn@mas.to

anchore

Agenda

00 Supply Chain Attacks

01 Log4Shell - What Actually Happened

02 Software Bill of Materials

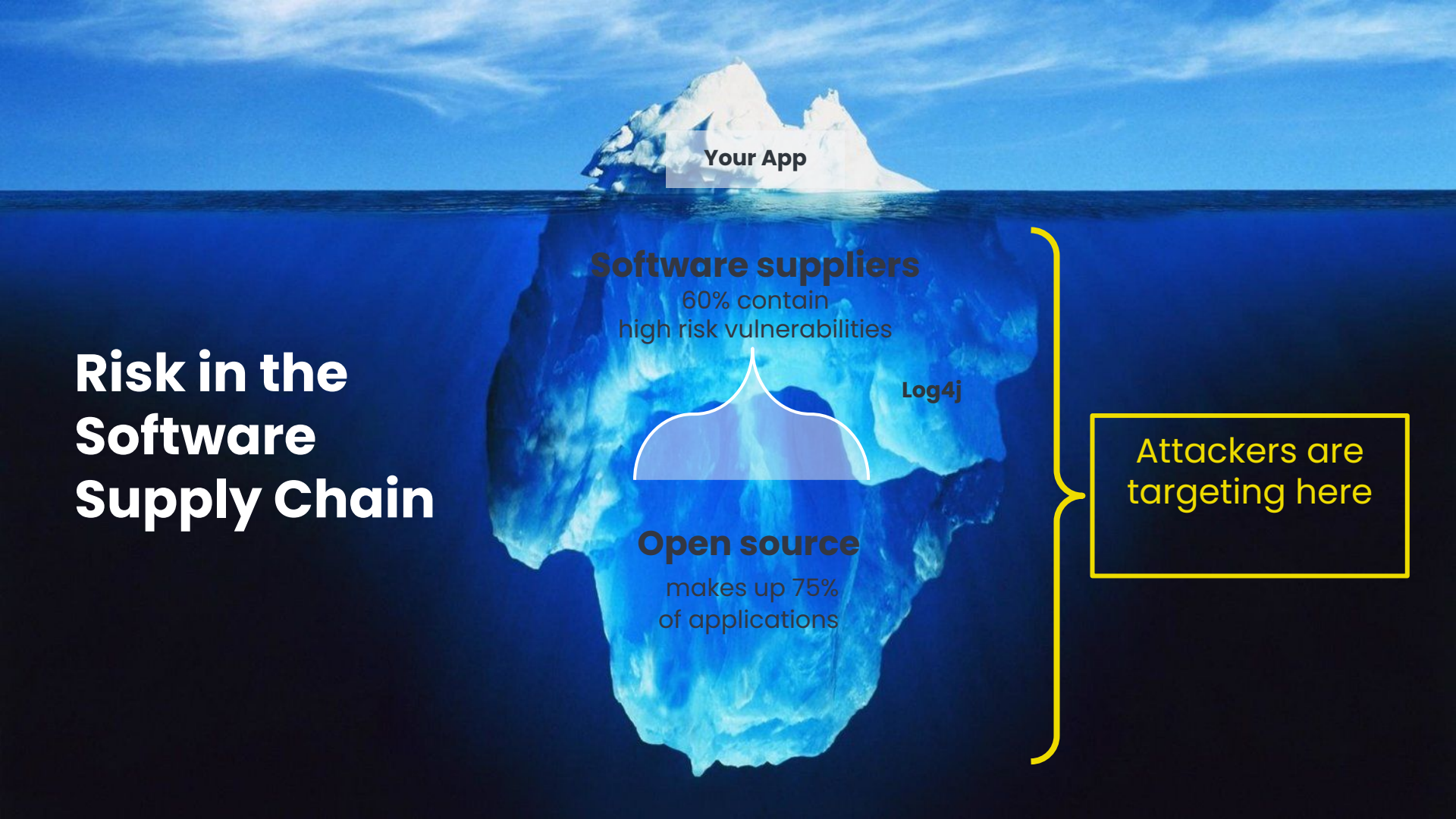
03 Now what?

anchore

anchore

What are Supply Chain Attacks?

Risk in the Software Supply Chain



Application Security | 4 MIN READ | NEWS

Supply Chain Attack Pushes Out Malware to More than 250 Media Websites

TA569 has modified the JavaScript of a legitimate content and advertising engine used by news affiliates, in order to spread the FakeUpdates initial access framework.

November 03, 2022

Popular Cryptocu
Exchange dYdX H
Its NPM Account I

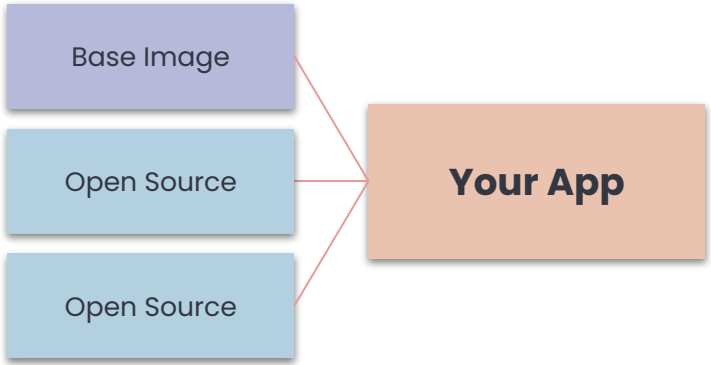
W4SP Stealer Stings Python Developers in Supply Chain Attack

Threat actors continue to push malicious Python packages to the popular PyPI service, striking with typosquatting, authentic sounding file names, and hidden imports to fool developers and steal their information.

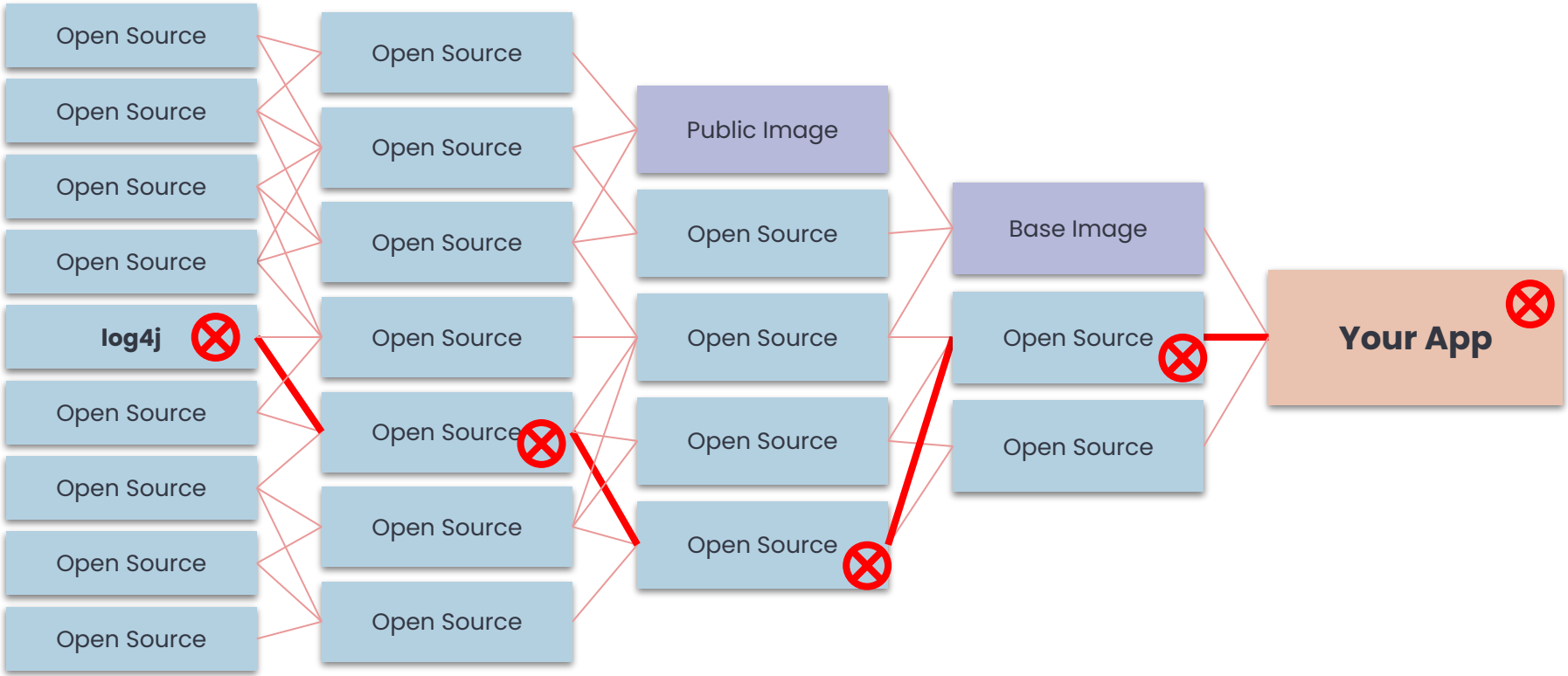
Robert Lemos
Contributing Writer, Dark Reading

November 04, 2022

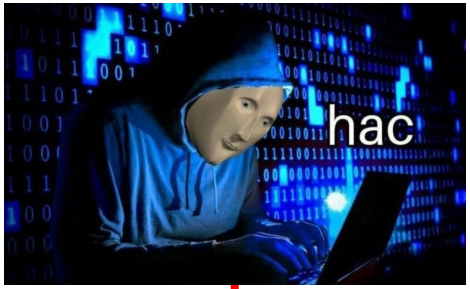
Software Supply Chain: The Iceberg



Software Supply Chain: The ~~Iceberg~~ Funnel



The Reverse Funnel



Compromised Package

Open Source
Open Source
Open Source

Public Image
Open Source
Open Source
Open Source
Open Source

Base Image
Open Source
Open Source
Open Source
Open Source
Open Source
Open Source

Docker Image
Commercial Software
Open Source
Open Source
Open Source
Open Source
Open Source
Docker Image
Open Source

anchore

anchore

Log4Shell Impact

What is log4j? What is log4shell?

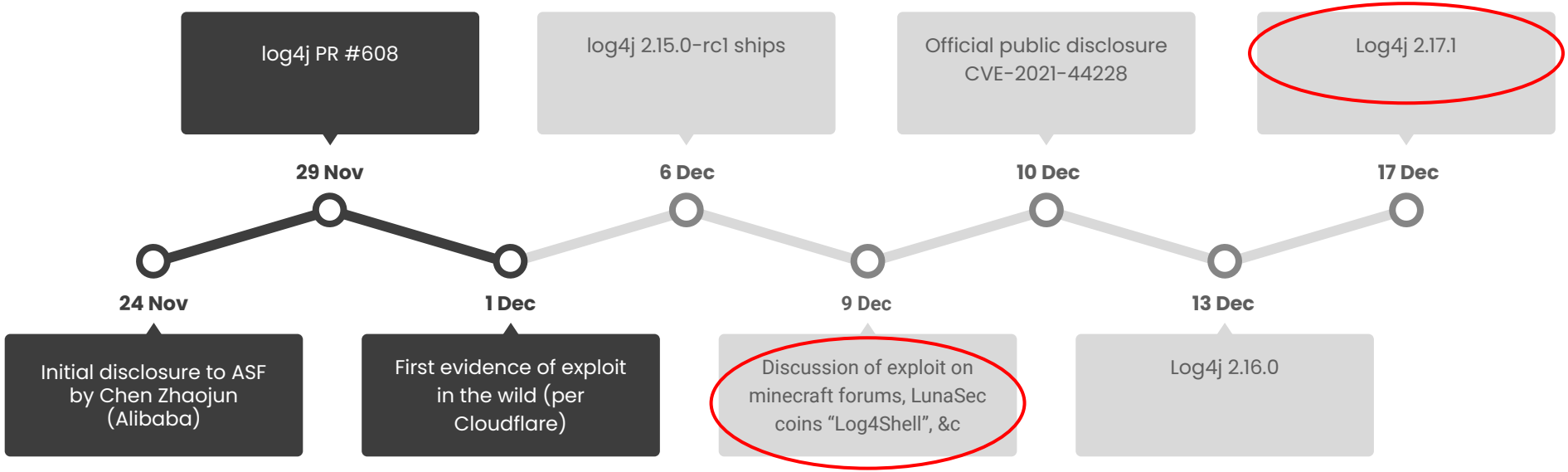
What is log4j? What is log4shell?

anchore

log4j Timeline - Background

- 2001 - Initial release
- 14 Sep 2013 - Vulnerability is introduced 2.0-beta9
- 3 Aug 2016 - Potential exploit presented at Black Hat

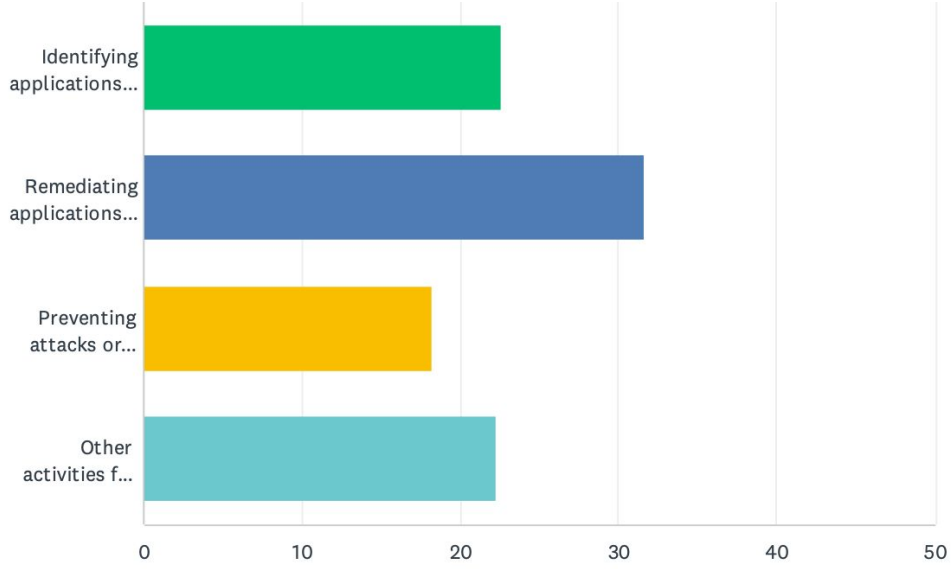
log4shell Timeline



anchore

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

Answered: 195 Skipped: 15



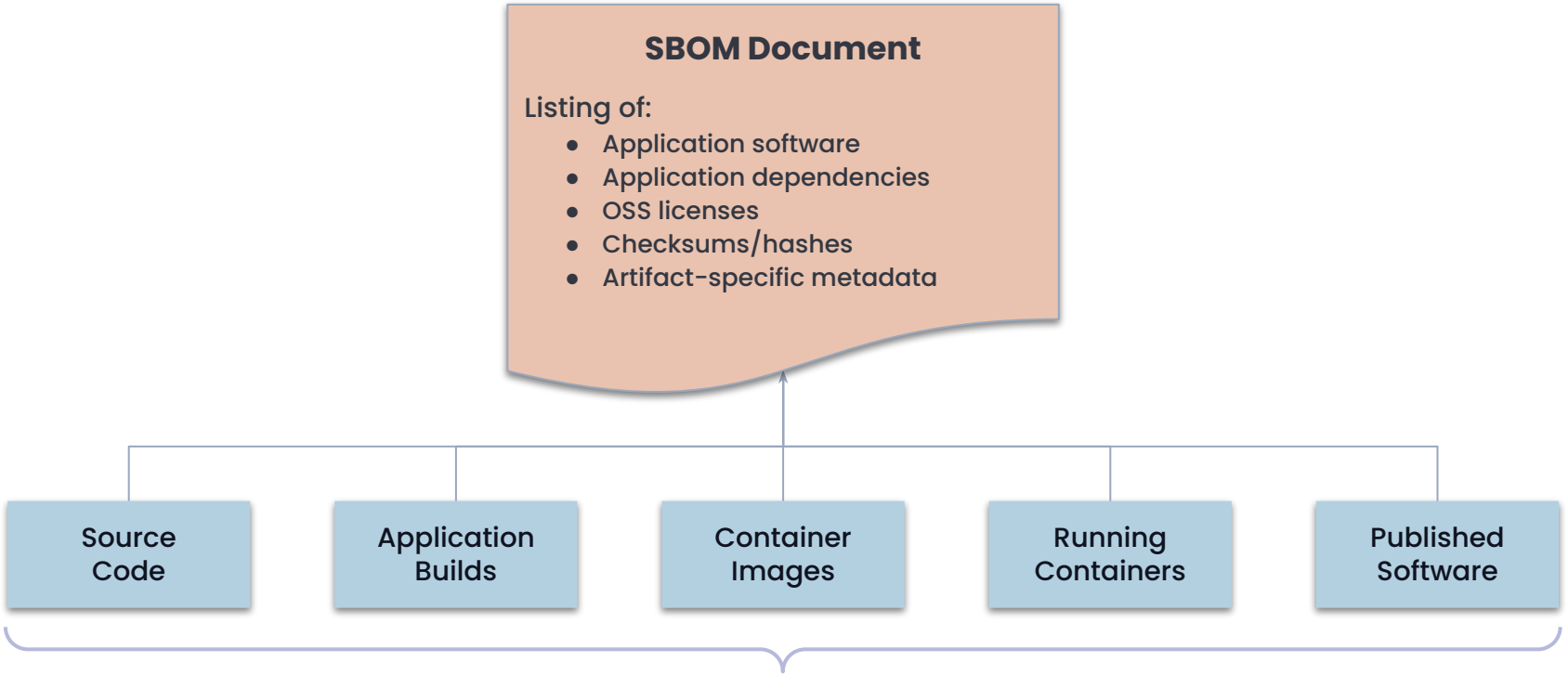
anchore

What is a Software Bill of Materials?

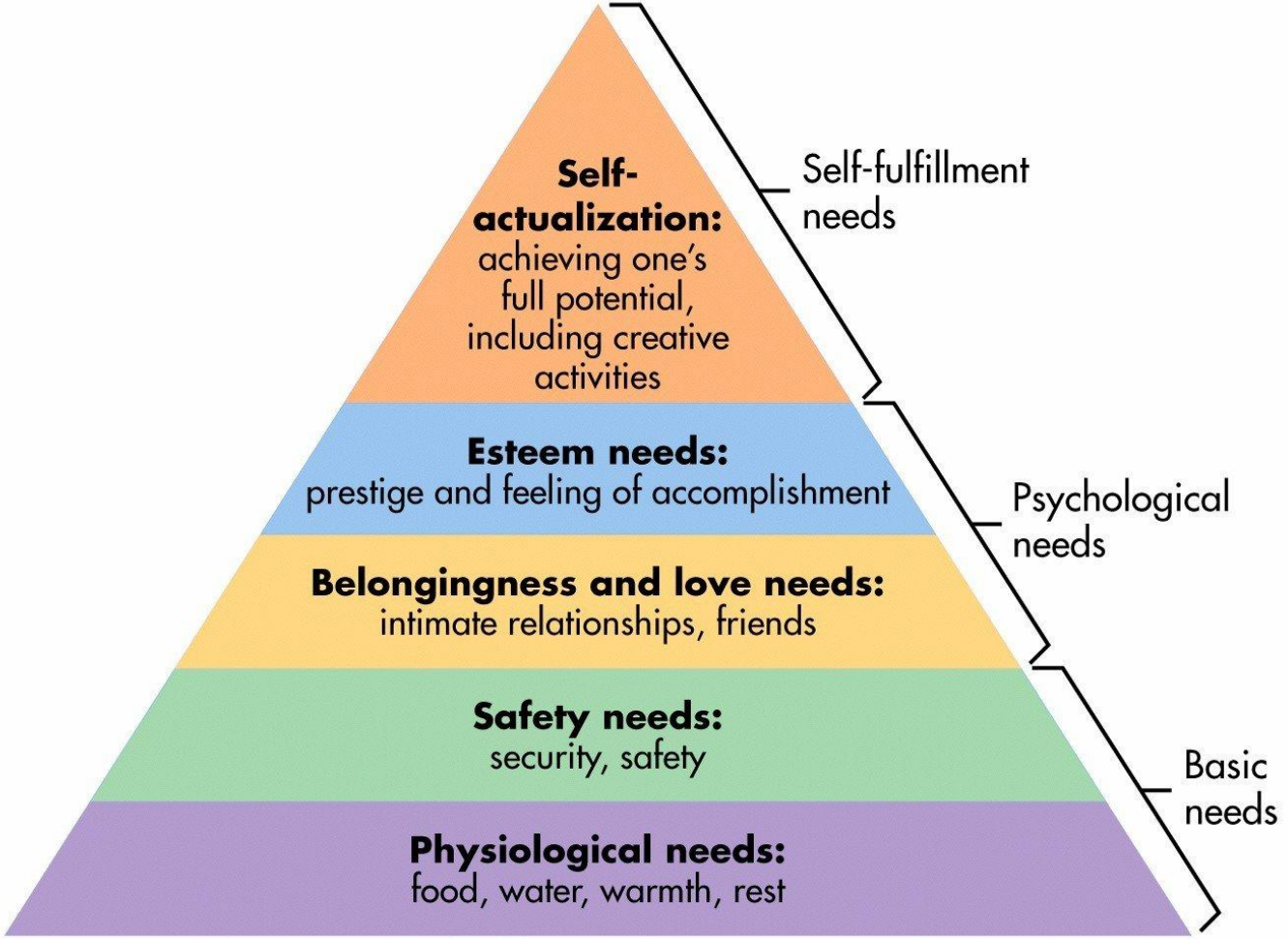
What is an SBOM?

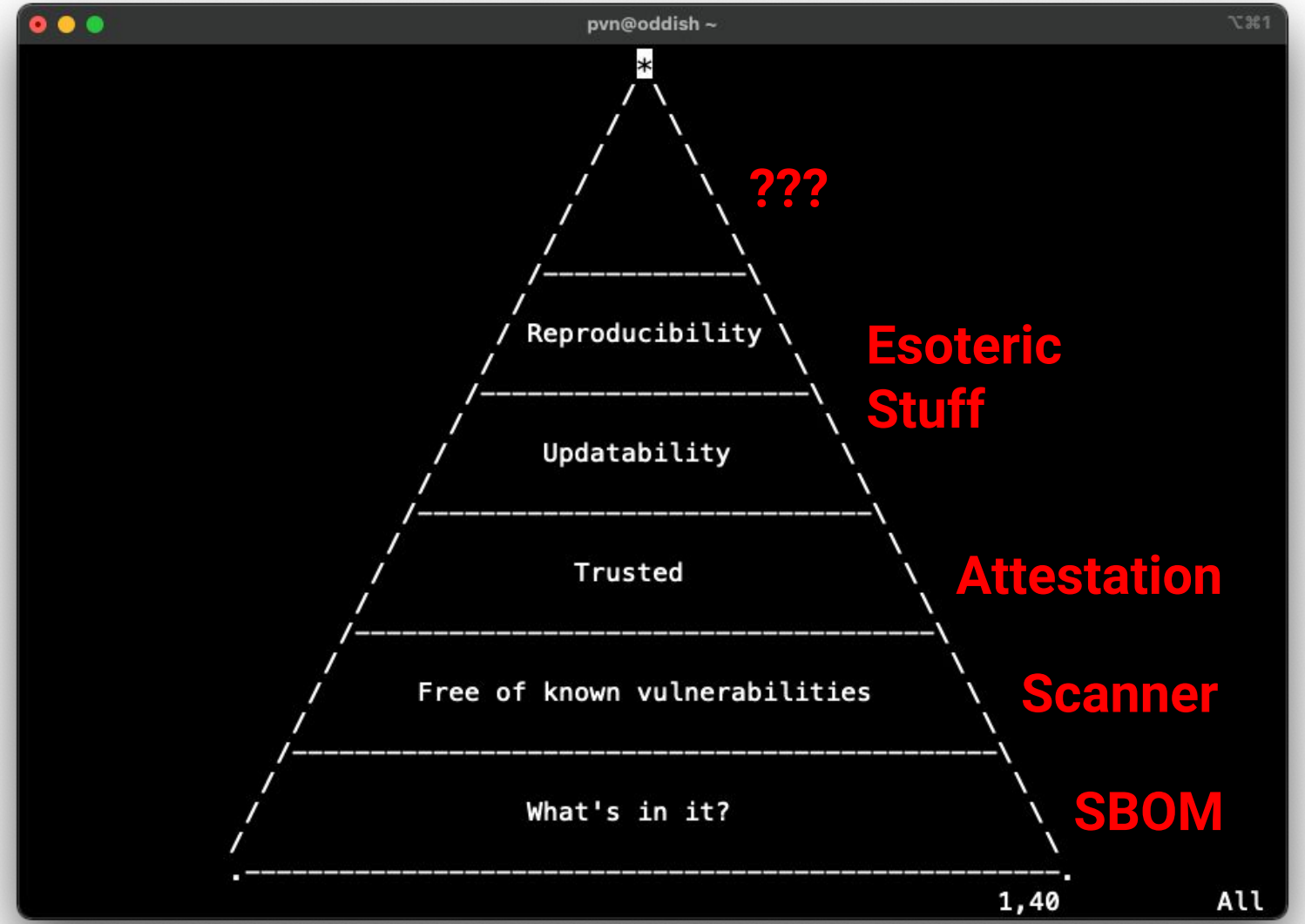
anchore

What is an SBOM?



Types of artifacts from which SBOMs can be generated


Self-actualization: achieving one's full potential, including creative activities
Esteem needs: prestige and feeling of accomplishment
Belongingness and love needs: intimate relationships, friends
Safety needs: security, safety
Physiological needs: food, water, warmth, rest
Self-fulfillment needs
Psychological needs
Basic needs


pvn@oddish ~
⌥⌘1
*
???
Reproducibility
Esoteric
Stuff
Updatability
Trusted
Attestation
Free of known vulnerabilities
Scanner
What's in it?
SBOM
1,40
All

SBOM: Current State

Obscured Dependencies

OpenSSL 3.x

???

Open Source

Node Base Image

dependencies

Your Code

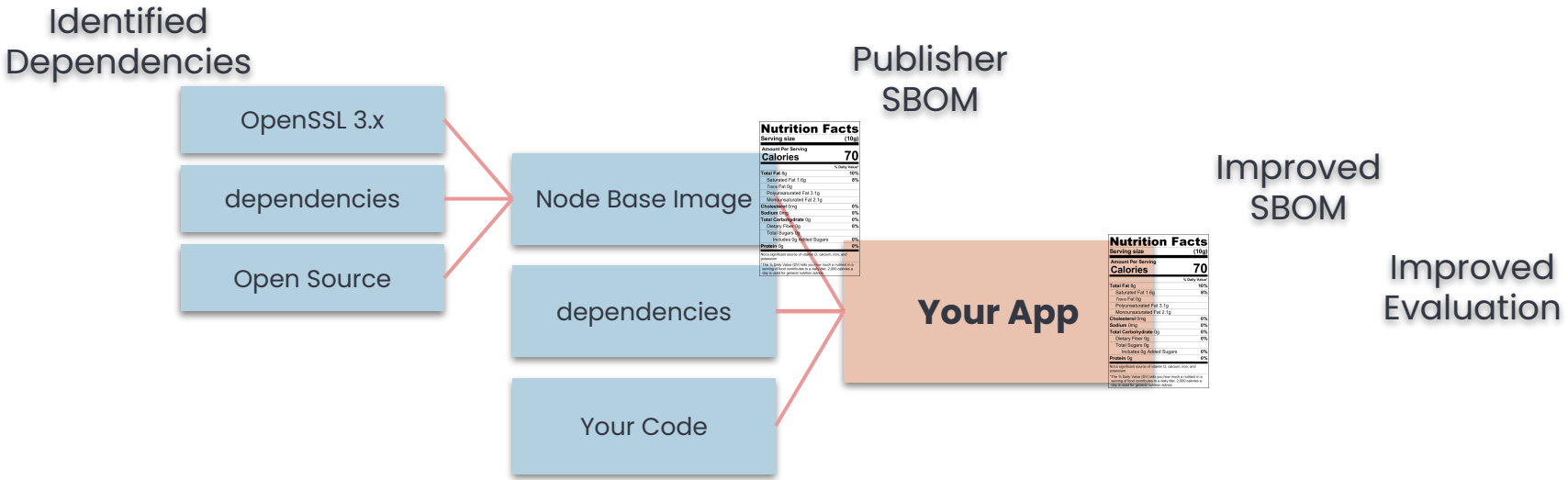
Your App

Imperfect SBOM

Imperfect Evaluation

anchore

SBOM: Better State



anchore

anchore

Now What?

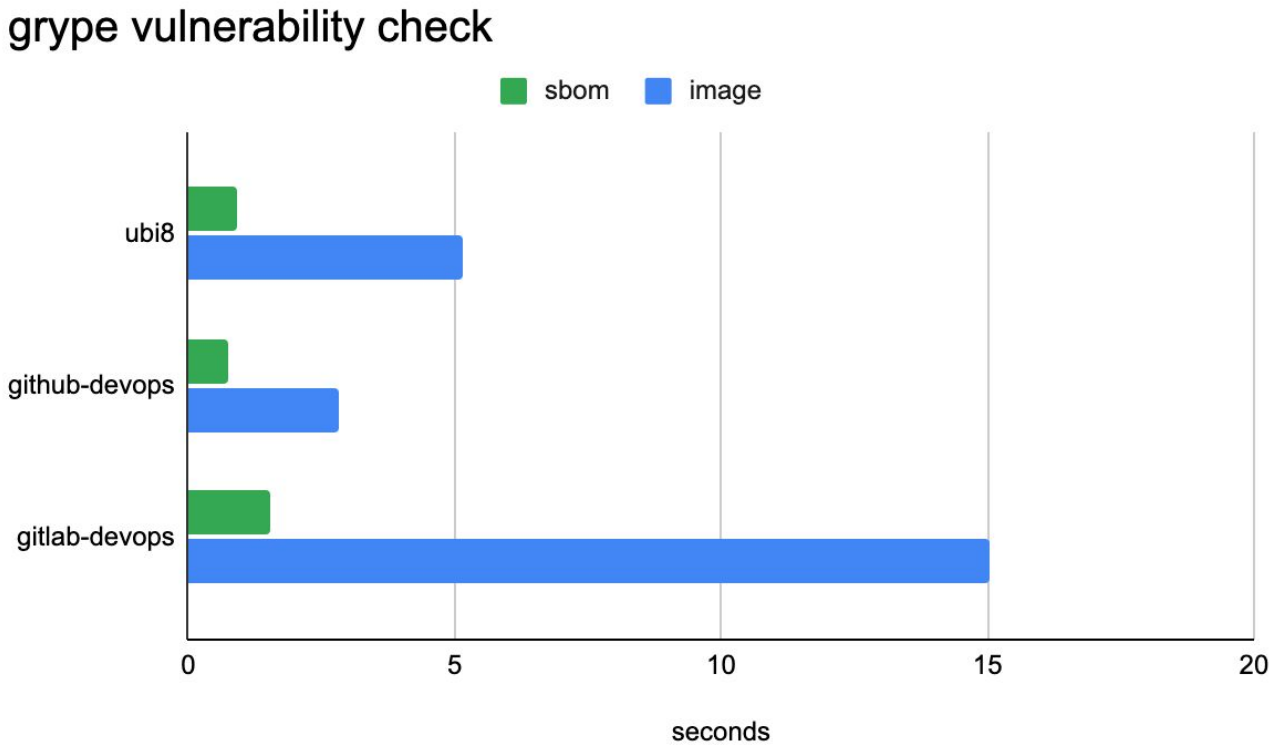
How will this improve my life?

How Do SBOMs Actually Help?

anchore

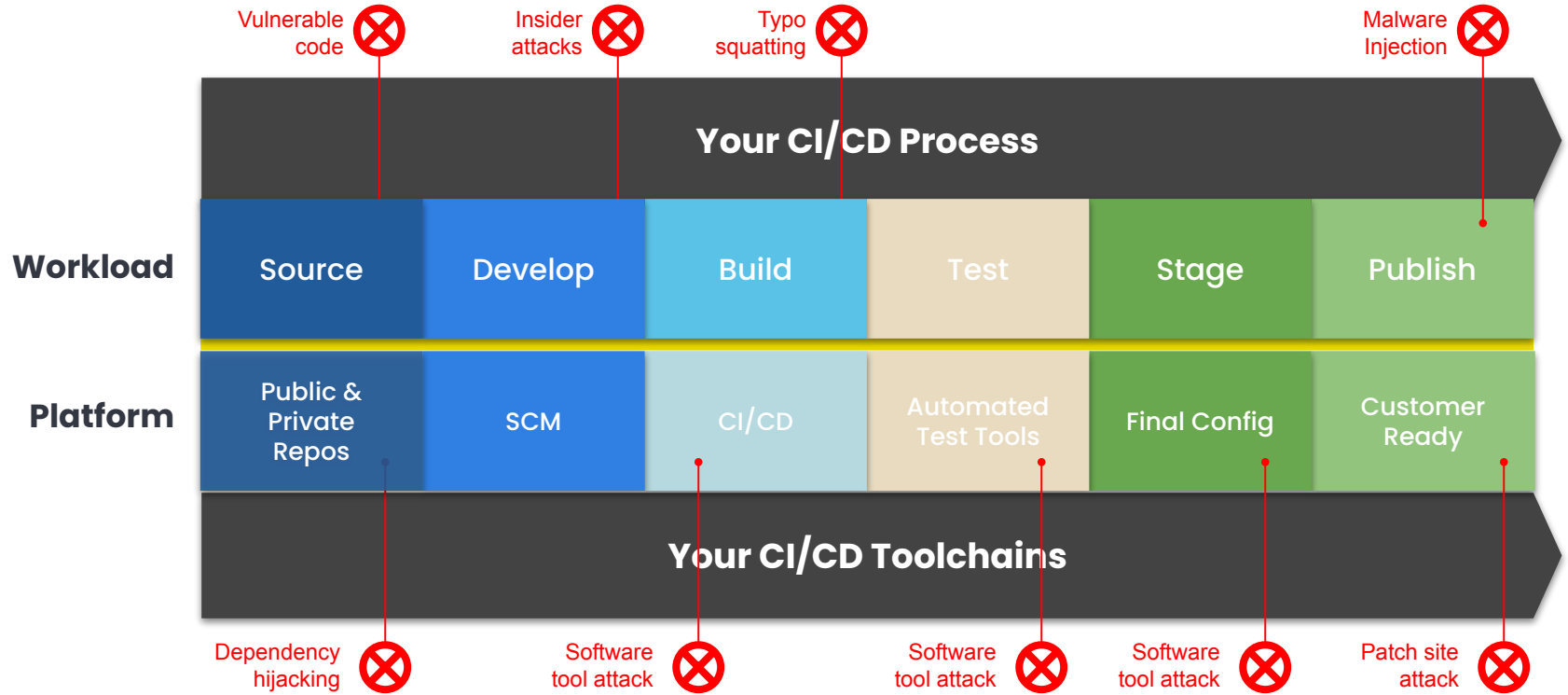
Grype Scan Timing

	ubi8	github-devops	gitlab-devops
sbom	0.917	0.751	1.55
image	5.159	2.839	15.031



anchore

SBOMs Enable Continuous Evaluation



anchore

Paul V. Novarese
@pvn

When I tweak the demo 10 minutes before my presentation

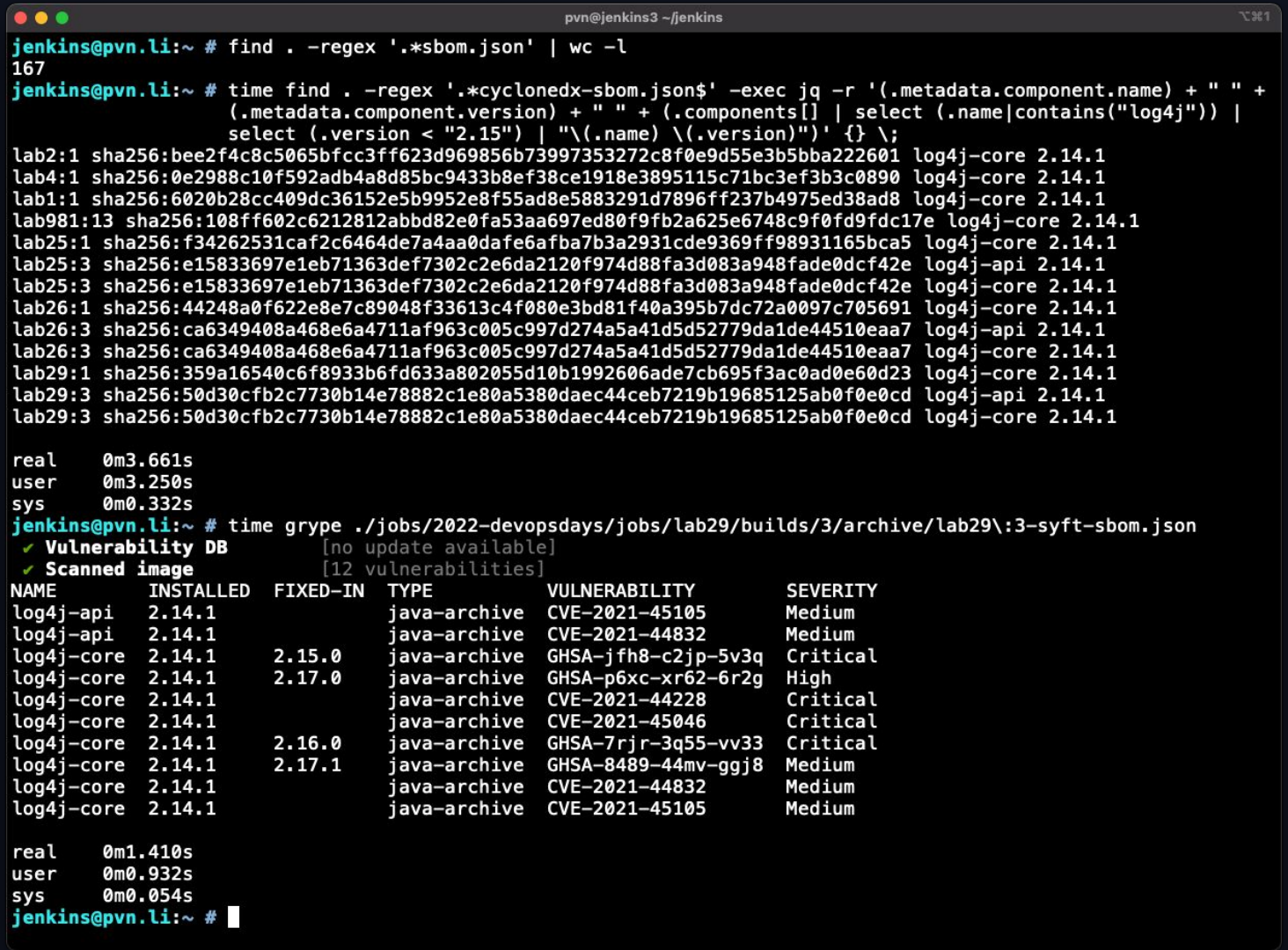
8:41 PM · Aug 26, 2018 · Twitter for iPhone

DIY Demos and Labs

- **Create SBOMs, Find log4j, Integrate with Jenkins (Difficulty: Easy)**
  - https://github.com/pvnovarese/2022-devopsdays
  - Includes instructions on deploying a disposable Jenkins container

- **Additional Labs (Difficulty: Medium)**
  - https://github.com/pvnovarese/2022-devopsworld
  - Additional Labs

- **GitHub SBOM Action:**
  - https://github.com/marketplace/actions/anchore-sbom-action


pvn@jenkins3 ~/jenkins
jenkins@pvn.li:~ # find . -regex '.*sbom.json' | wc -l
167
jenkins@pvn.li:~ # time find . -regex '.*cyclonedx-sbom.json$' -exec jq -r '(.metadata.component.name) + " " + (.metadata.component.version) + " " + (.components[] | select (.name|contains("log4j")) | select (.version < "2.15") | "\(.name) \(.version)")' {} \;
lab2:1 sha256:bee2f4c8c5065bfcc3ff623d969856b73997353272c8f0e9d55e3b5bba222601 log4j-core 2.14.1
lab4:1 sha256:0e2988c10f592adb4a8d85bc9433b8ef38ce1918e3895115c71bc3ef3b3c0890 log4j-core 2.14.1
lab1:1 sha256:6020b28cc409dc36152e5b9952e8f55ad8e5883291d7896ff237b4975ed38ad8 log4j-core 2.14.1
lab981:13 sha256:108ff602c6212812abbd82e0fa53aa697ed80f9fb2a625e6748c9f0fd9fdc17e log4j-core 2.14.1
lab25:1 sha256:f34262531caf2c6464de7a4aa0dafe6afba7b3a2931cde9369ff98931165bca5 log4j-core 2.14.1
lab25:3 sha256:e15833697e1eb71363def7302c2e6da2120f974d88fa3d083a948fade0dcf42e log4j-api 2.14.1
lab25:3 sha256:e15833697e1eb71363def7302c2e6da2120f974d88fa3d083a948fade0dcf42e log4j-core 2.14.1
lab26:1 sha256:44248a0f622e8e7c89048f33613c4f080e3bd81f40a395b7dc72a0097c705691 log4j-core 2.14.1
lab26:3 sha256:ca6349408a468e6a4711af963c005c997d274a5a41d5d52779da1de44510eaa7 log4j-api 2.14.1
lab26:3 sha256:ca6349408a468e6a4711af963c005c997d274a5a41d5d52779da1de44510eaa7 log4j-core 2.14.1
lab29:1 sha256:359a16540c6f8933b6fd633a802055d10b1992606ade7cb695f3ac0ad0e60d23 log4j-core 2.14.1
lab29:3 sha256:50d30cfb2c7730b14e78882c1e80a5380daec44ceb7219b19685125ab0f0e0cd log4j-api 2.14.1
lab29:3 sha256:50d30cfb2c7730b14e78882c1e80a5380daec44ceb7219b19685125ab0f0e0cd log4j-core 2.14.1
real 0m3.661s
user 0m3.250s
sys 0m0.332s
jenkins@pvn.li:~ # time grype ./jobs/2022-devopsdays/jobs/lab29/builds/3/archive/lab29\:3-syft-sbom.json
✔ Vulnerability DB [no update available]
✔ Scanned image [12 vulnerabilities]
NAME INSTALLED FIXED-IN TYPE VULNERABILITY SEVERITY
log4j-api 2.14.1 java-archive CVE-2021-45105 Medium
log4j-api 2.14.1 java-archive CVE-2021-44832 Medium
log4j-core 2.14.1 2.15.0 java-archive GHSA-jfh8-c2jp-5v3q Critical
log4j-core 2.14.1 2.17.0 java-archive GHSA-p6xc-xr62-6r2g High
log4j-core 2.14.1 java-archive CVE-2021-44228 Critical
log4j-core 2.14.1 java-archive CVE-2021-45046 Critical
log4j-core 2.14.1 2.16.0 java-archive GHSA-7rjr-3q55-vv33 Critical
log4j-core 2.14.1 2.17.1 java-archive GHSA-8489-44mv-ggj8 Medium
log4j-core 2.14.1 java-archive CVE-2021-44832 Medium
log4j-core 2.14.1 java-archive CVE-2021-45105 Medium
real 0m1.410s
user 0m0.932s
sys 0m0.054s
jenkins@pvn.li:~ #

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

anchore

anchore

References &c

Notes, Additional Reading, Bibliography

Log4j Bibliography &c

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 Reading List

Draft of upcoming site content for SBOM.me: https://github.com/joshbressers/sbom-examples/blob/readme-update/site/index.md

Making Better SBOMs: https://kccncna2022.sched.com/event/182GT/ – https://www.youtube.com/watch?v=earq775L4fc

Announcing GUAC: https://security.googleblog.com/2022/10/announcing-guac-great-pairing-with-slsa.html

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

Solar Winds post mortem: https://www.lawfareblog.com/solarwinds-and-holiday-bear-campaign-case-study-classroom

Profound Podcast - Episode 10 (John Willis and Josh Corman):
https://www.buzzsprout.com/1758599/8761108-profound-dr-deming-episode-10-josh-corman-captain-america

Creating a trusted container supply chain: https://thenewstack.io/creating-a-trusted-container-supply-chain/

Footnotes

Other notes:
Slide 6: https://www.mend.io/resources/blog/popular-cryptocurrency-exchange-dydx-has-had-its-npm-account-hacked/
Slide 6: https://www.mend.io/resources/blog/cybercriminals-targeted-users-of-packages-with-a-total-of-1-5-billion-weekly-downloads-on-npm/
Slide 6: https://www.darkreading.com/threat-intelligence/w4sp-stealer-aims-to-sting-python-developers-in-supply-chain-attack
Slide 6: https://www.darkreading.com/application-security/supply-chain-attack-pushes-out-malware-to-more-than-250-media-websites
Slide 12: https://www.blackhat.com/docs/us-16/materials/us-16-Munoz-A-Journey-From-JNDI-LDAP-Manipulation-To-RCE.pdf
Slide 13: https://hypixel.net/threads/psa-there-is-a-fatal-remote-code-execution-exploit-in-minecraft-and-its-by-typing-in-chat.4703238/
Slide 13: https://twitter.com/_r_netsec/status/1469120458083962882
Slide 13: https://twitter.com/eastdakota/status/1469800951351427073
Slide 19: Maslow's Hierarchy of Supply Chain Needs: https://www.youtube.com/watch?v=rcP8QHFMwCw
Slide 20: https://kccncna2022.sched.com/event/182GT/ – https://www.youtube.com/watch?v=earq775L4fc

Images used for SBOM generation timing benchmarks:

- registry.access.redhat.com/ubi8:latest
- https://gitlab.com/pvn_test_images/devops-supply-chain
- https://github.com/pvnovarese/devops-supply-chain-demo

Integration of cosign with syft: https://github.com/anchore/syft/issues/510
Add support for hints in syft: https://github.com/anchore/syft/issues/31

Best Practices for Securing the Software Supply Chain

00 Centralized, secure CI/CD process for all software

01 Build images from trusted sources

02 Automate security testing and policy enforcement

03 Deploy only trusted images into production

anchore

anchore

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/2022-devopsdays

anchore

Appendix A

SBOM Formats

Existing SBOM formats

	SPDX "Software Package Data eXchange"	**CycloneDX**	**SWID** "Software ID"
Organization	SPDX Workgroup (~20 orgs) under the Linux Foundation	A "meritocratic, consensus-based community project" with a Industry Working Group	ISO/IEC Joint Technical Committee Trusted Computing Group
Initial Draft	2010	2017	2009
Formats	RDF, XLS, SPDX, YAML, JSON	XML, JSON	XML, CBOR (CoSWID only)
Spec	spdx.github.io/spdx-spec BS ISO/IEC 5962 - 2020 Draft	github.com/CycloneDX/specification	iso.org/standard/65666.html ISO/IEC 19770-2:2015

Existing SBOM formats: Use Cases

	SPDX "Software Package Data eXchange"	**CycloneDX**	**SWID** "Software ID"
Original use cases	License management	For use with OWASP Dependency-Track	Inventory and change tracking
Unique Features	Extensive support for expressing license details	Extensible format and integrates SPDX license IDs, pURL, and other external identifiers	Deeply integrated into the build and publishing process for a software component
Use cases of latest format versions	• Tracking attributes of multiple software components (e.g. vendor, license, version, etc.) • Generically describe packages, containers, os distributions, archives, etc • Integrity verification of software components and sub-components		

A “good” SBOM describes...

What is being catalogued	Each item uniquely	What did the cataloguing
For example a running system, a machine image, a container image, etc.	Such as each component name, version, UUID, and relationships to other components.	The tool that generated the document with its configuration.

A “great” SBOM also includes...

In scope and out of scope

For example “only these paths were searched” or “only JARS and RPMs are being search for”.

Exceptional conditions

Such as warnings or errors that occur during processing or missing environmental factors

Additional metadata

Such as Java pom properties, key-values, additional RPM DB tag entries, and licenses.

anchore

Introducing Syft

- Syft is an **open source tool that generates SBOMs** from container images and filesystems

- Syft supports **many package ecosystems:**
  - APK, DEB, RPM, Ruby Bundle, Python Wheel/Egg/requirements.txt, JavaScript NPM/Yarn, Java JAR/EAR/WAR, Jenkins plugin JPI/HPI, Go modules, Rust Crate

- Syft also supports **multiple output formats**
  - Syft-Native
  - CycloneDX
  - SPDX

anchore

Appendix B

Additional Bonus Slides

Paul Novarese
Software Supply Chain Security at Anchore
9mo · Edited

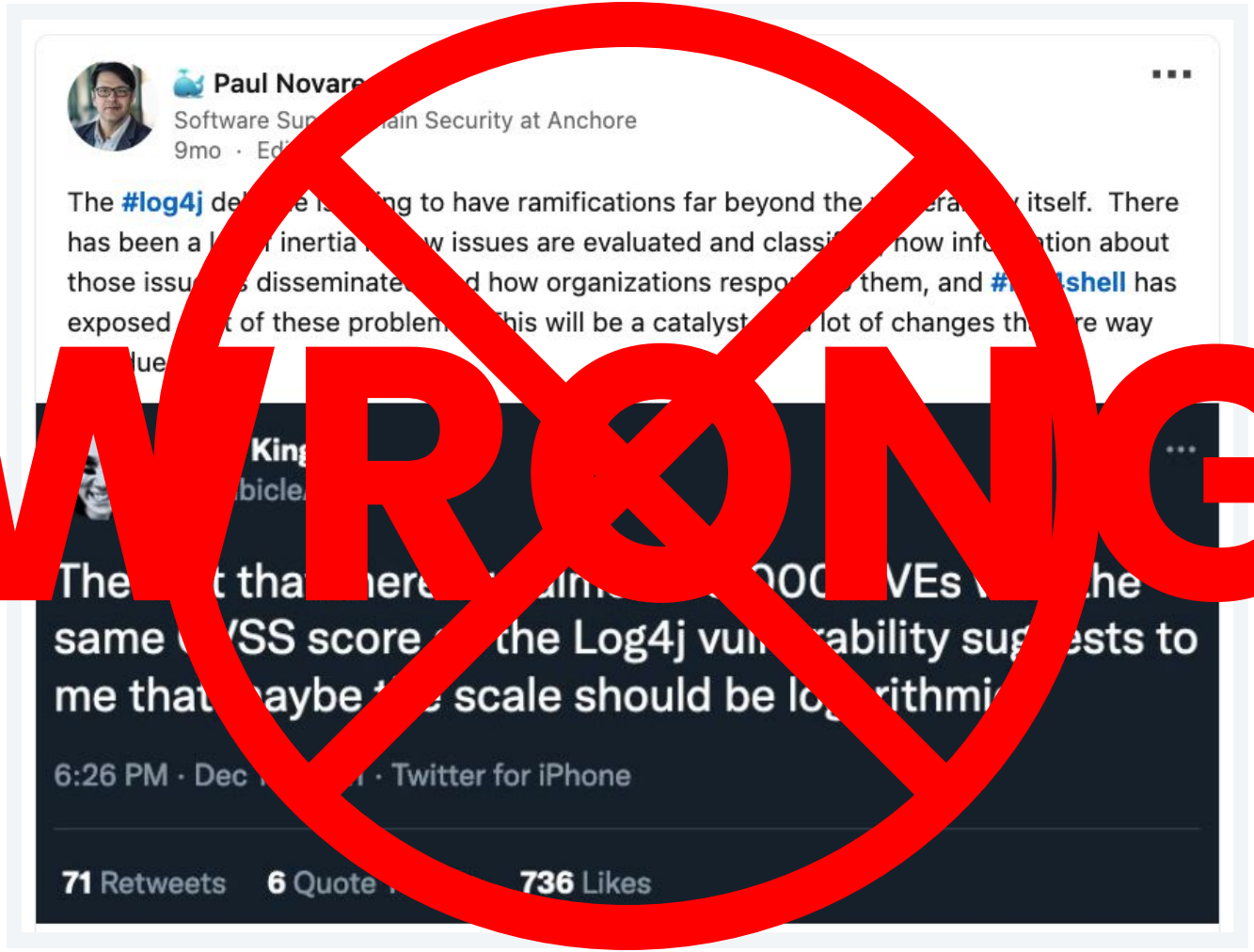
The #log4j debacle is going to have ramifications far beyond the vulnerability itself. There has been a lot of inertia in how issues are evaluated and classified, how information about those issues is disseminated, and how organizations respond to them, and #log4shell has exposed a lot of these problems. 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

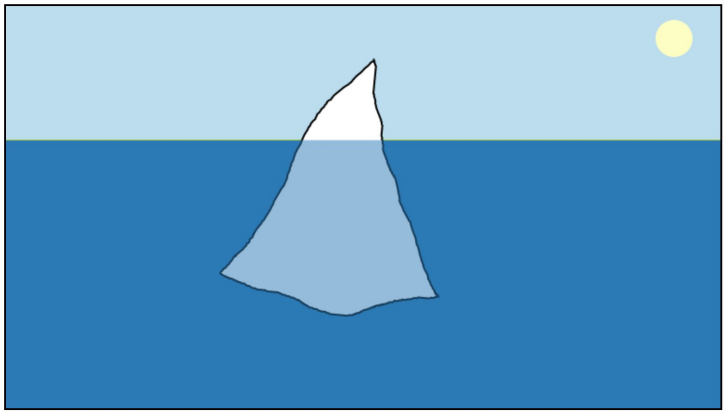

Paul Novare
Software Sup ain Security at Anchore
9mo · Ed
The #log4j de g to have ramifications far beyond the era itself. There
has been a l inertia ow issues are evaluated and classi how info tion about
those issu disseminate d how organizations respo them, and # shell has
exposed t of these problem his will be a catalyst lot of changes th re way
lue
King
bicle
The t tha here aim 000 VEs the
same VSS score the Log4j vul rability sug ests to
me that aybe scale should be lo rithmi
6:26 PM · Dec · Twitter for iPhone
71 Retweets 6 Quote 736 Likes
WRONG

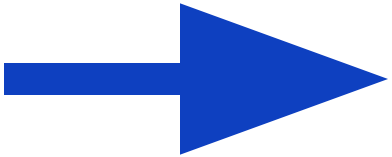
Quick Aside

Iceberger

Draw an iceberg and see how it will float.

(Inspired by a tweet by @GlacialMeg)

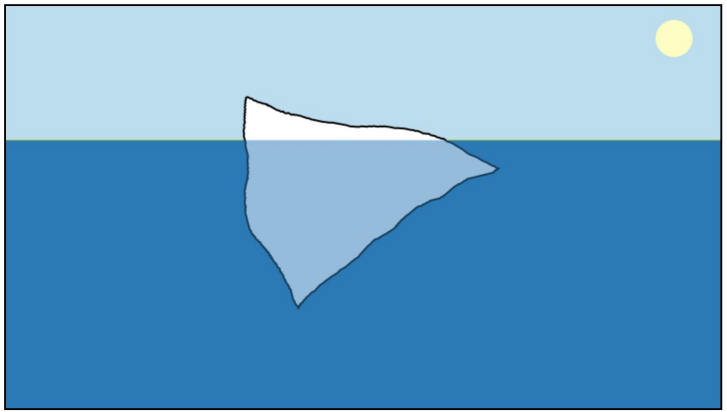




Iceberger

Draw an iceberg and see how it will float.

(Inspired by a tweet by @GlacialMeg)



Jake Williams
@MalwareJake

Hey infosec: remember that your job is risk reduction, not risk elimination. There's a BIG difference.

9:31 PM · Aug 29, 2021 · Twitter for Android

258 Retweets **26** Quote Tweets **1,677** Likes

Dan Kaminsky
@dakami

Amateurs think about vulnerabilities, professionals think about vectors.

4:04 PM · Aug 12, 2017 · Twitter Web Client