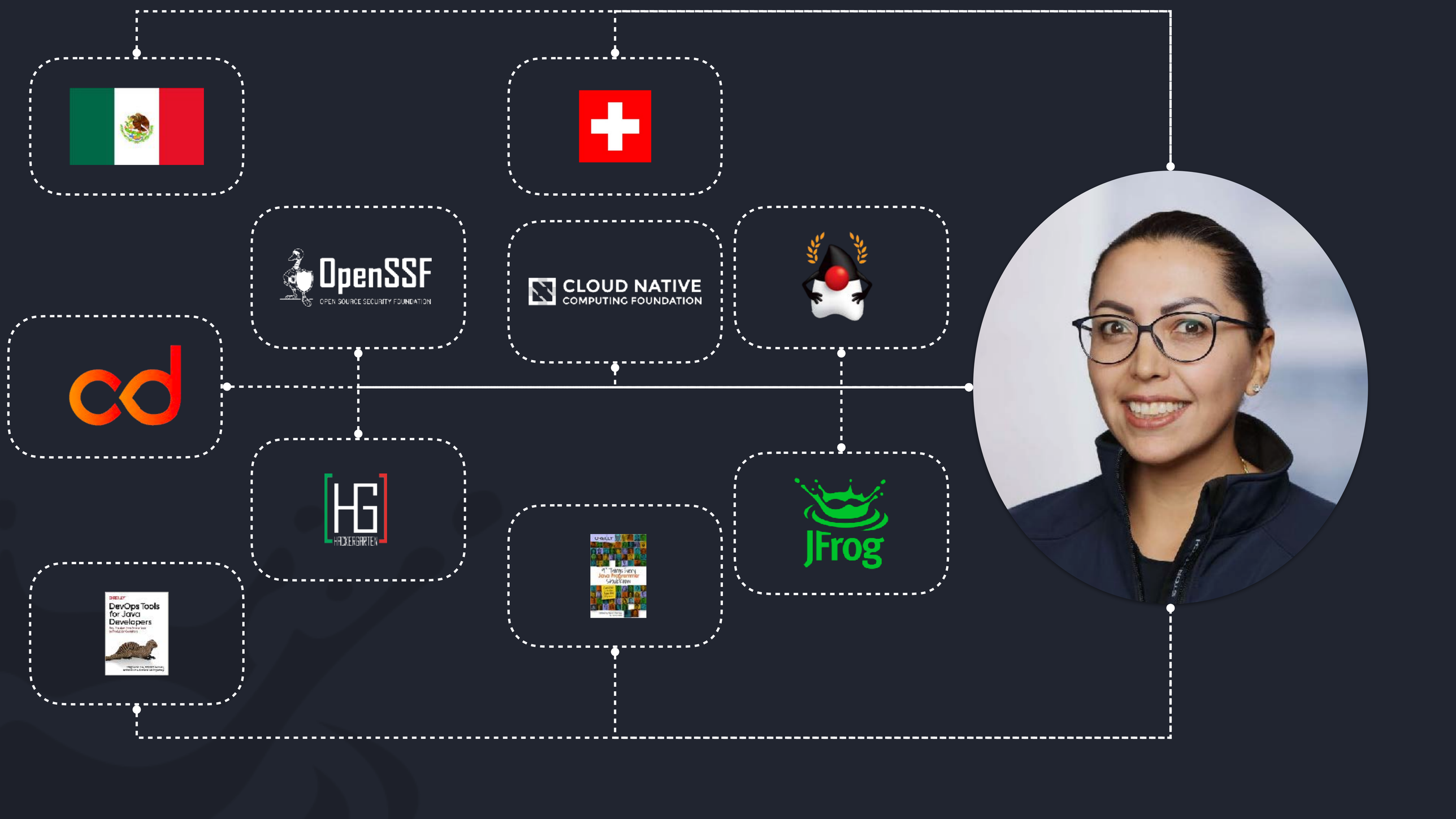
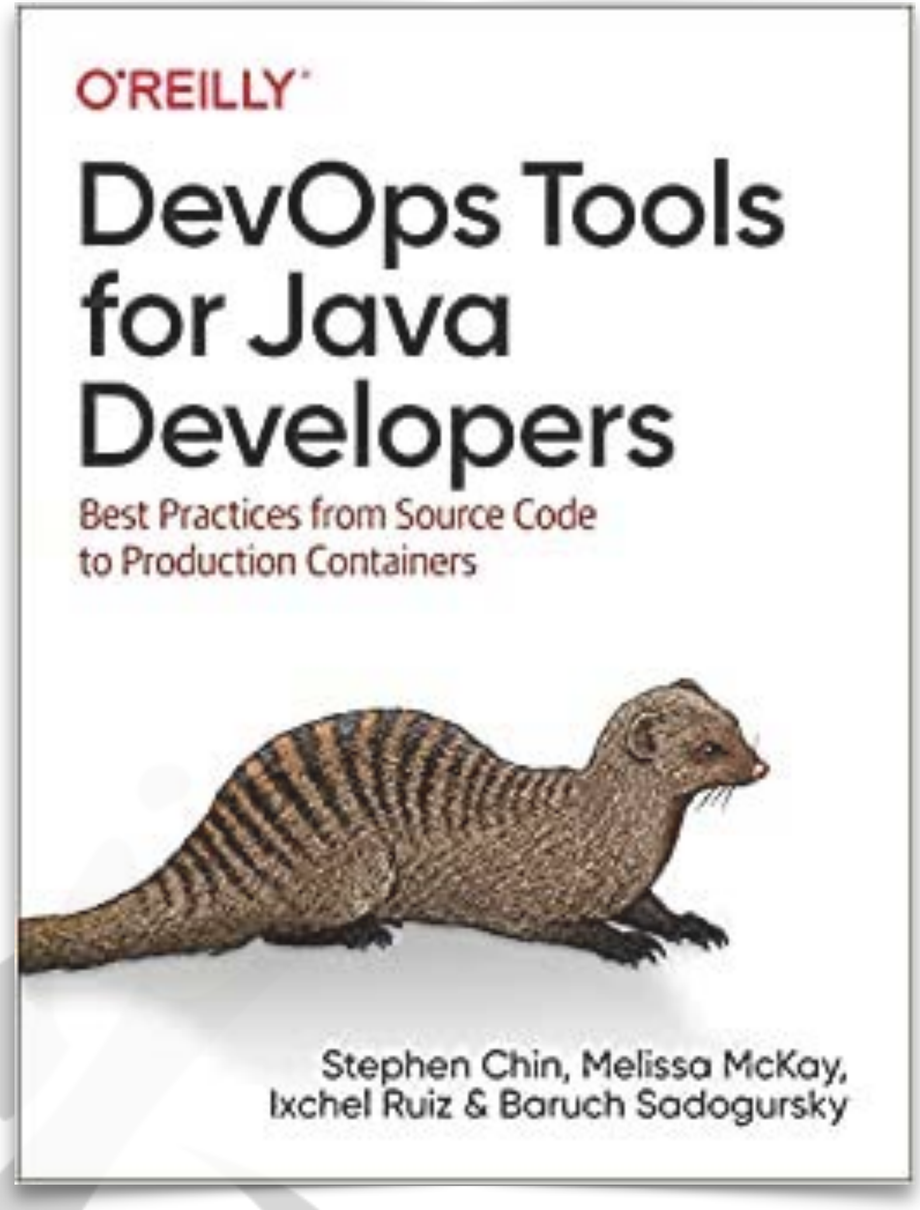
@Conf42: DevSecOps 2022

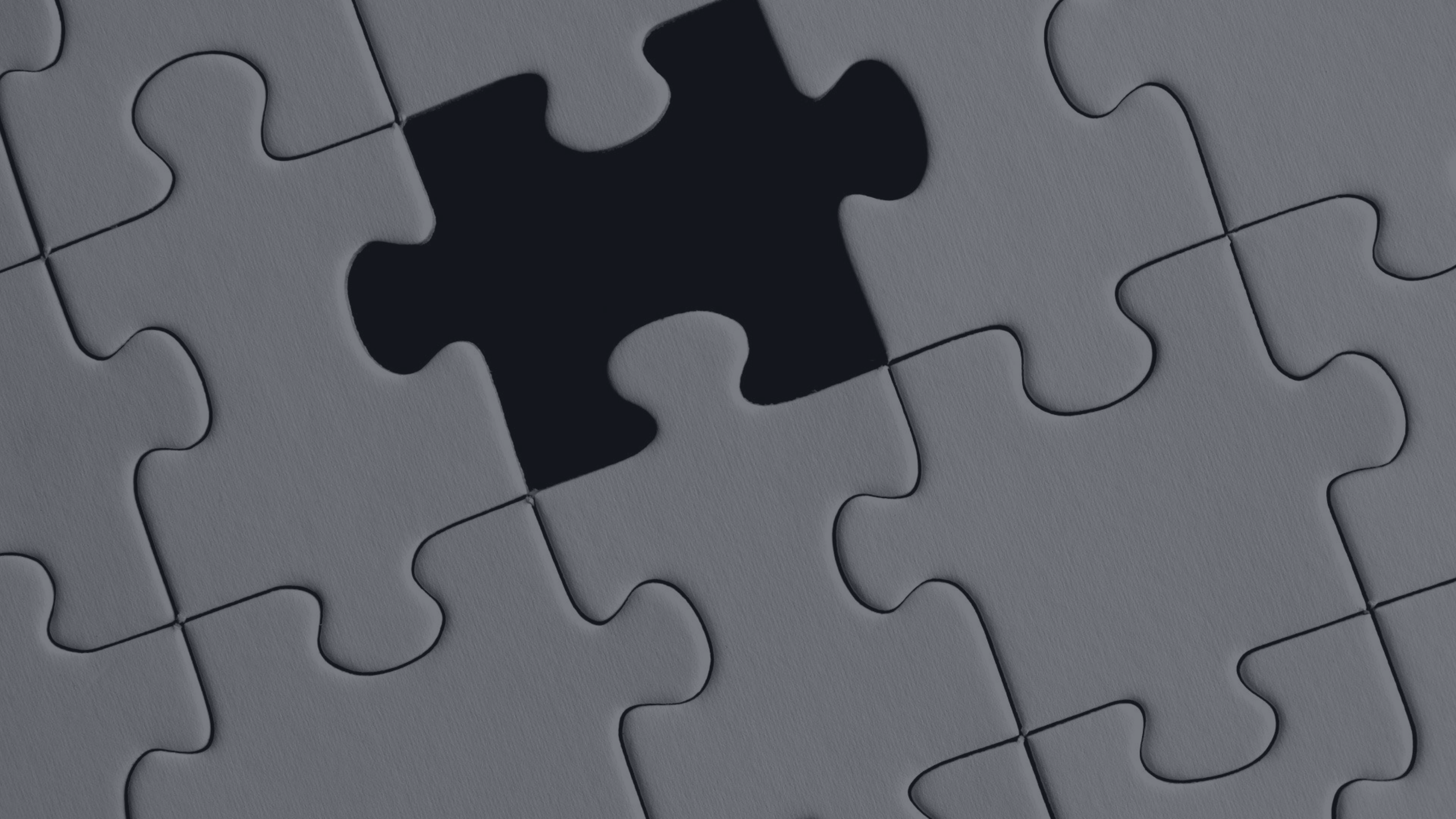
A journey of the thousand binaries

Ixchel Ruiz


OpenSSF
OPEN SOURCE SECURITY FOUNDATION
CLOUD NATIVE
COMPUTING FOUNDATION
cd
HG
HACKERGARTEN
JFrog
DevOps Tools
for Java
Developers

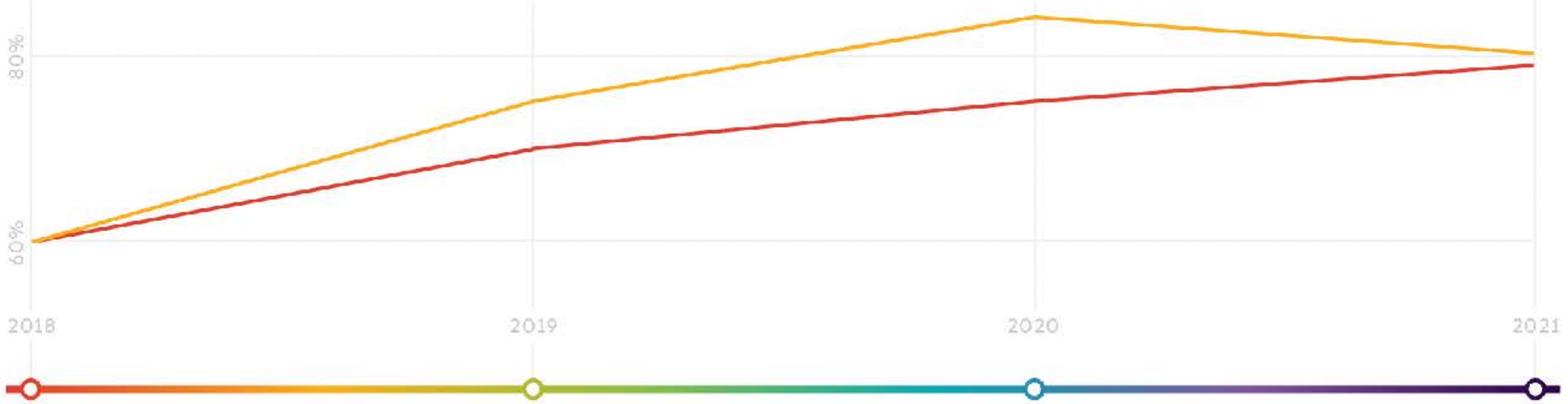

O'REILLY
DevOps Tools for Java Developers
Best Practices from Source Code to Production Containers
Stephen Chin, Melissa McKay, Ixchel Ruiz & Baruch Sadogursky


“World runs on software …”



78% OF CODE IN CODEBASES WAS OPEN SOURCE

81% CONTAINED AT LEAST ONE VULNERABILITY



88% CONTAINED COMPONENTS THAT HAD NO NEW DEVELOPMENT IN TWO YEARS

85% CONTAINED OPEN SOURCE THAT WAS MORE THAN FOUR YEARS OUT-OF-DATE

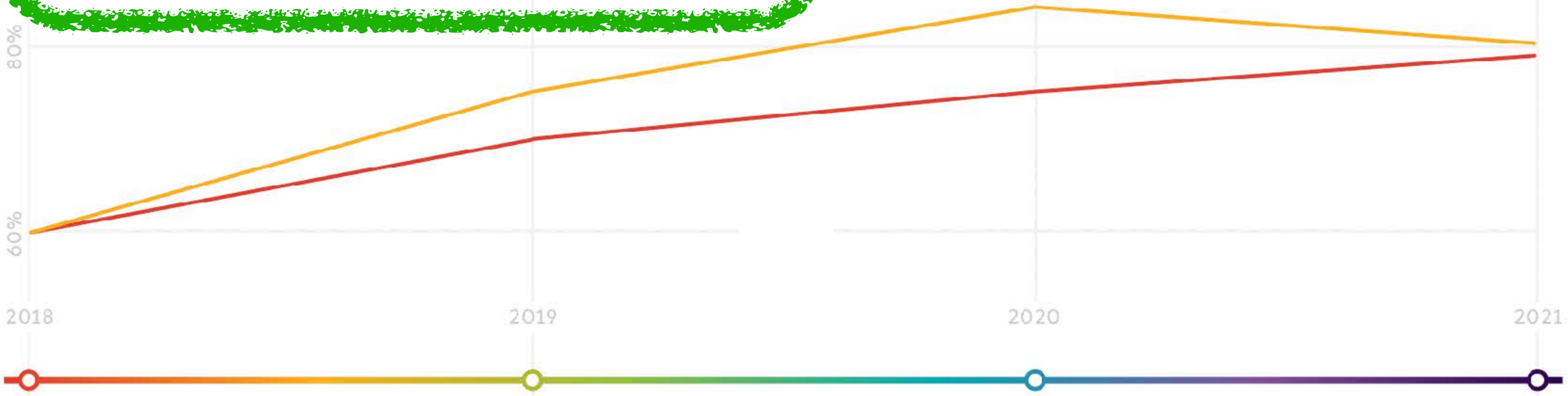
OF AUDITED CODEBASES HAD LICENSE CONFLICTS

CONTAINED OPEN SOURCE WITH NO LICENSE OR CUSTOM LICENSE

UTILIZED COMPONENTS THAT WERE NOT THE LATEST VERSION

78% OF CODE IN CODEBASES WAS OPEN SOURCE

81% CONTAINED AT LEAST ONE VULNERABILITY



88% CONTAINED COMPONENTS THAT HAD NO NEW DEVELOPMENT IN TWO YEARS

85% CONTAINED OPEN SOURCE THAT WAS MORE THAN FOUR YEARS OUT-OF-DATE

Dependencies

Not all are the same.

True? Or False?

“Dependencies are collections containing *high-quality tested* code that provides functionality that required significant *expertise* to develop. ”

“Dependency managers like NPM have made possible that *trivial* functionality can be packaged and published”

Dependencies

Types of Dependencies

Frameworks, libraries, packages, modules and resources.

Types of Dependencies

“Collection of files for example templates, media (audio, video or images), plain text files or blobs that need to be included by applications to execute correctly”

Resources

“Collection of files for example templates, media (audio, video or images), plain text files or blobs that need to be included by applications to execute correctly”

Resources

“*Set* of methods and functions that provide a self contained functionality. A module usually has an interface that specifies both the functionality it provides as well as the functionality it depends on”

Module

“*Set* of methods and functions that provide a self contained functionality. A module usually has an interface that specifies both the functionality it provides as well as the functionality it depends on”

Module

“A *collection* of modules that hold in general the same functional purpose. Usually a directory that contains a file that describe metadata about the package.”

Package

“A *collection* of modules that hold in general the same functional purpose. Usually a directory that contains a file that describe metadata about the package.”

Package

“A *collection* of *related* functionality defined in several packages, is essentially a set of functions that you can call, each call does some work and returns control to the client or application that executed said function. ”

Library

“A *collection* of *related* functionality defined in several packages, is essentially a set of functions that you can call, each call does some work and returns control to the client or application that executed said function. ”

Library

“A framework embodies some abstract design, with more behaviour built in. In order to use it you need to insert your behaviour into various places in the framework .The framework's code then calls your code at these points.”

Frameworks

“A framework embodies some abstract design, with more behaviour built in. In order to use it you need to insert your behaviour into various places in the framework .The framework's code then calls your code at these points.”

Frameworks

Frameworks — Platforms

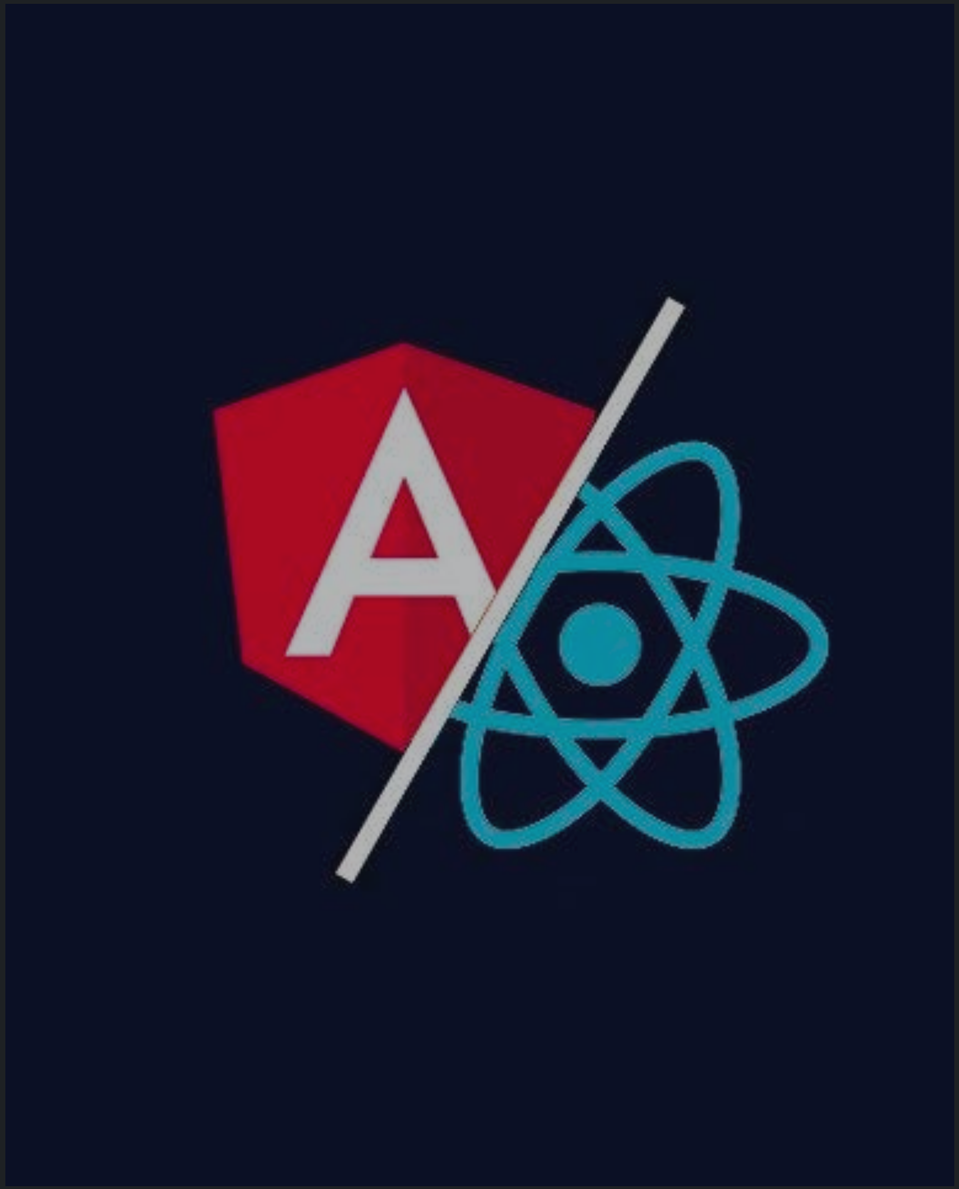
Key points

- Functionality
- LOC (size)
- Opinionated
- Integration between functional components
- Roadmap — Versioning
- Licensing
- Tests

Angular — React

Framework — Platform

Library



github.com/parro-it/awesome-micro-npm-packages

Product Team Enterprise Explore Marketplace Pricing

Search / Sign in Sign

parro-it / awesome-micro-npm-packages Public

Notifications Fork 444 Star 4.2k

Code Issues Pull requests Actions Projects Wiki Security Insights

master 2 branches 0 tags Go to file Code

parro-it Merge pull request #70 from TomerAberbach/patch-1 ... a1ec75e on 15 Sep 2021 146 commits

LICENSE	Create LICENSE	5 years ago
code-of-conduct.md	Create code-of-conduct.md	6 years ago
contributing.md	Update broken link to `amending-a-commit-guide.md`	4 years ago
readme.md	docs: add quetie to new data structures section	6 months ago

readme.md

Awesome Micro npm Packages awesome

> A curated list of small, focused Node.js modules.

Inspired by the awesome list thing.

Articles

- One-line node modules
- Build small single purpose modules
- Module best practices
- Evaluating Packages Part 1 - Turn to community

About

A curated list of small, focused npm packages.

Readme
MIT License
Code of conduct
4.2k stars
197 watching
444 forks

Releases

No releases published

Packages

No packages published

Contributors 39

Perspective

“Cadence of update, migrations costs or cleanup efforts”

Key considerations

What can possibly go wrong?

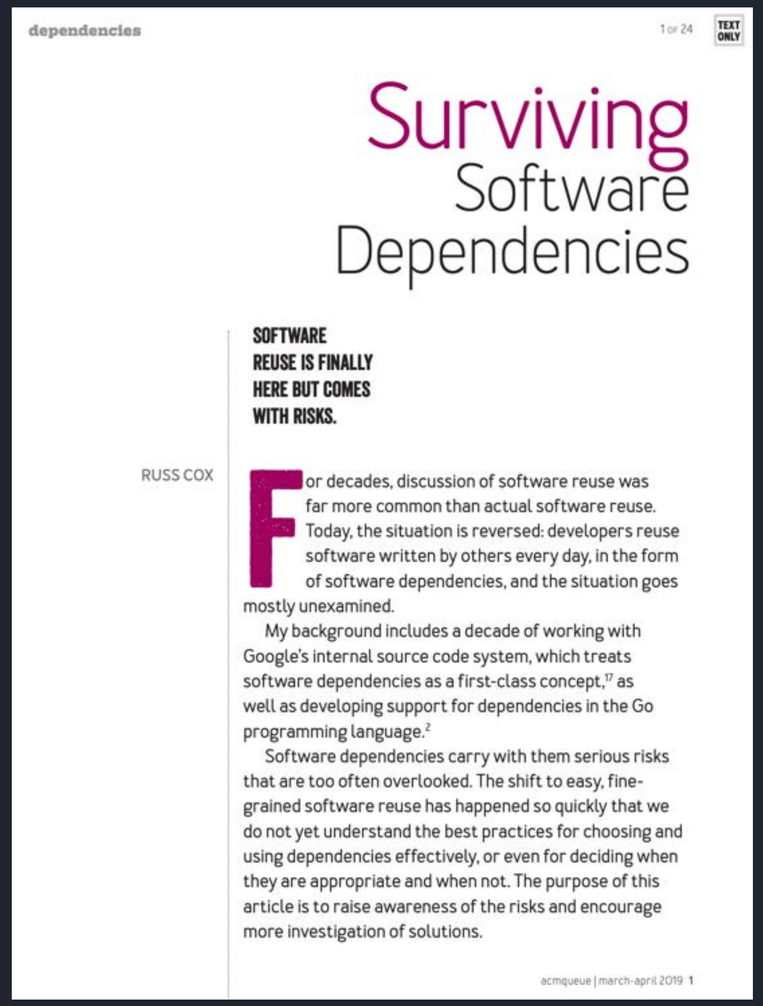
“ Adding a dependency outsources the work of developing that code—designing, writing, testing, debugging, and maintaining—to someone else”

The unknown programmer

Surviving Software Dependencies

Russ Cox

acmqueue



dependencies 1 of 24 TEXT ONLY

Surviving Software Dependencies

SOFTWARE REUSE IS FINALLY HERE BUT COMES WITH RISKS.

RUSS COX

For decades, discussion of software reuse was far more common than actual software reuse. Today, the situation is reversed: developers reuse software written by others every day, in the form of software dependencies, and the situation goes mostly unexamined.

My background includes a decade of working with Google's internal source code system, which treats software dependencies as a first-class concept,[17] as well as developing support for dependencies in the Go programming language.[2]

Software dependencies carry with them serious risks that are too often overlooked. The shift to easy, fine-grained software reuse has happened so quickly that we do not yet understand the best practices for choosing and using dependencies effectively, or even for deciding when they are appropriate and when not. The purpose of this article is to raise awareness of the risks and encourage more investigation of solutions.

acmqueue | march-april 2019 1

$$\text{expectedCost} = \sum_{b} \text{cost}(b) * \text{probability}(b)$$

Cost of adopting a bad dependency

Inspect the Dependency

“Is the documentation clear? Does the API have a clear design? ”

Inspect Dependency: Design

“Is the code well written? Does it look like the authors have been careful, conscientious, and consistent? Does it look like code you would want to debug?”

Inspect Dependency: Code Quality

“Does the code have tests? Can you run them? Do they pass? Tests establish that the code's basic functionality is correct”

Inspect Dependency: Testing

“Issue tracker. Are there many open bug reports? How long have they been open? Are there many fixed bugs? Have any bugs been fixed recently?”

Inspect Dependency: Bug fixing

“How long has the code been actively maintained? Is it actively maintained now? How many people work on the package?”

Inspect Dependency: Maintenance

“Do many other packages depend on this code? How often others write about using the project?”

Inspect Dependency: Usage

“Will you be processing untrusted inputs with the package? If so, does it seem to be robust against malicious inputs? Does it have a history of security problems listed in the NVD (National Vulnerability Database)?”

Inspect Dependency: Security

“Is the code properly licensed? Does it have a license at all? Is the license acceptable for your project or company?”

Inspect Dependency: Licensing

“Does the code have dependencies of its own? List all the transitive dependencies”

Inspect Dependency: Dependencies

Dependencies

Tools

JFrog

Products Solutions Developers Resources Pricing

En START FOR FREE

SOFTWARE COMPOSITION ANALYSIS WITH AGILITY

JFrog Xray is an **application security SCA tool** that integrates security directly into your DevOps workflows, enabling you to deliver trusted software releases faster.

JFrog Xray **fortifies your software supply chain** and scans your entire pipeline from Git to your IDE, through your CI/CD Tools, and all the way through distribution to deployment.

Xray is being used as a security solution to assist us in finding out which docker images that are published out to our artifactory instance are vulnerable, and digging down into all the different layers within those docker images and finding out exactly what needs to be fixed.

BRAD BECKTELL, DEVOPS ENGINEER, KROGER

Kroger

JFrog XRAY

https://jfrog.com/xray/

OWASP Dependency-Track

Main | Features | Integrations | Installation | News | Our Supporters | Executive Order 14028

For more details about Dependency-Track see the projects website at dependencytrack.org

Dependency-Track is an intelligent Component Analysis platform that allows organizations to identify and reduce risk in the software supply chain. Dependency-Track takes a unique and highly beneficial approach by leveraging the capabilities of Software Bill of Materials (SBOM). This approach provides capabilities that traditional Software Composition Analysis (SCA) solutions cannot achieve.

https://owasp.org/www-project-dependency-track/

ossf / wg-best-practices-os-developers Public

generated from ossf/project-template

Watch 52 | Fork 32 | Starred 298

Code | Issues 9 | Pull requests 1 | Discussions | Actions | Projects | Security | Insights

main | wg-best-practices-os-developers / docs / Concise-Guide-for-Evaluating-Open-Source-Software.md | Go to file

david-a-wheeler Tweak GitHub Markdown URLs with #readme ... Latest commit 2371a97 16 days ago History

2 contributors

49 lines (43 sloc) | 6.65 KB

Raw | Blame

Concise Guide for Evaluating Open Source Software 2022-09-01

As a software developer, before using open source software (OSS) dependencies or tools, identify candidates and evaluate the leading ones against your needs. To evaluate a potential OSS dependency for security and sustainability, consider these questions (all tools or services listed are merely examples):

1. **Can you avoid adding it?** Can you use an existing (possibly indirect) dependency instead? Every new dependency increases the attack surface (a subversion of the new dependency, or its transitive dependencies, may subvert the system).
2. **Are you evaluating the intended version?** Ensure you are evaluating the intended version of the software, not a personal fork nor an attacker-controlled fork. These techniques help to counter the common "typosquatting" attack (where an attacker creates an "almost-correct" name).
    i. Check its name and the project website for the link.
    ii. Verify the fork relation on GitHub/GitLab.

https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Evaluating-Open-Source-Software.md#readme

ossf / scorecard Public

Watch 46 | Fork 221 | Starred 2.5k

Code | Issues 168 | Pull requests 8 | Discussions | Actions | Projects | Wiki | Security | Insights

main | 13 branches | 19 tags | Go to file | Add file | Code

naveensrinivasan Ignore mock clients for code coverage ... e303a1b yesterday 1,206 commits

.github	Update CODEOWNERS (#1701)	8 days ago
artwork	Add and use compressed Scorecard logos (#1492)	2 months ago
checker	Fix cmd panic (#1692)	9 days ago
checks	Unit tests for pinned_dependencies	yesterday
clients	allow empty committer (#1714)	3 days ago
cloudbuild	Fix `scorecard version` in Scorecard Docker images (#1480)	2 months ago
cmd	cmd: Allow new scorecard to be instantiated with options (#1703)	8 days ago
cron	Fix golangci-lint issues	19 days ago
docs	Adding missing documentation for Token-Permissions (#1656)	13 days ago
e2e	Mark `License`, `Security-Policy` as commit-based (#1711)	6 days ago
errors	Only run allowed checks in different modes (#1579)	last month
log	Make verbosity levels case insensitive (#1650)	22 days ago
options	cmd: Allow new scorecard to be instantiated with options (#1703)	8 days ago
pkg	Add score to SARIF for all results (#1694)	8 days ago

About

Security Scorecards - Security health metrics for Open Source

scorecard | security-scorecards

Readme
Apache-2.0 License
Code of conduct
2.5k stars
46 watching
221 forks

Releases 18

v4.1.0 Latest
17 days ago

+ 17 releases

Packages 1

scorecard/gitcache

https://github.com/ossf/scorecard

ossf / wg-best-practices-os-developers Public

generated from ossf/project-template

Watch 52 | Fork 32 | Starred 298

Code | Issues 9 | Pull requests 1 | Discussions | Actions | Projects | Security | Insights

main | wg-best-practices-os-developers / docs / Concise-Guide-for-Developing-More-Secure-Software.md | Go to file

david-a-wheeler Tweak GitHub Markdown URLs with #readme ... Latest commit 2371a97 16 days ago History

5 contributors

29 lines (27 sloc) | 6.22 KB

Raw | Blame

Concise Guide for Developing More Secure Software 2022-09-01

Here is a concise guide for all software developers for software development, building, and distribution. All tools or services listed are merely examples.

1. **Ensure all privileged developers use multi-factor authentication (MFA) tokens**. This includes those with commit or accept privileges. MFA hinders attackers from "taking over" these accounts.
2. **Learn about secure software development.** Take, e.g., the free OpenSSF course or the hands-on Security Knowledge Framework course. SAFECode's Fundamental Practices for Secure Software Development provides a helpful summary.
3. **Use a combination of tools in your CI pipeline to detect vulnerabilities**. See the OpenSSF guide to security tools. Tools shouldn't be the *only* mechanism, but they scale.
4. **Evaluate software before selecting it as a direct dependency**. Only add it if needed, evaluate it (see Concise Guide for Evaluating Open Source Software), double-check its name (to counter typosquatting), and ensure it's retrieved from the correct repository.
5. **Use package managers**. Use package managers (system, language-level, and/or container-level) to automatically manage dependencies and enable rapid updates.
6. **Implement automated tests**. Include negative tests (tests that what shouldn't happen doesn't happen) and ensure the test suite is thorough enough to "ship if it passes the tests".
7. **Monitor known vulnerabilities in your software's direct & indirect dependencies**. E.g., enable basic scanning via GitHub's dependabot

https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Developing-More-Secure-Software.md#readme

Type Dependency Level Quality

Dependencies Coordinates


250MM
LUX-TOOLS
10mm

ONE tool at the centre

Universal Artifact Management

JFrog
ARTIFACTORY

JFrog
XRAY

https://jfrog.com/start-free/

JFROG ARTIFACTORY: THE DATABASE OF DEVOPS

JFrog XRAY

JFrog ARTIFACTORY

The Liquid Software Company

Frogbot

Test passing | GitHub Action Test passing | coverage 43% | go report A+

Table of contents

🤖 What is Frogbot?

Frogbot is a Git bot that does the following:

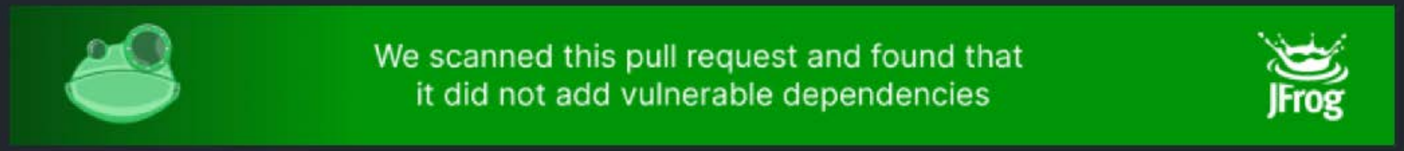
1. Scans pull requests for security vulnerabilities.
2. Opens pull requests with fixes for security vulnerabilities.

https://github.com/jfrog/frogbot

Pull request comments

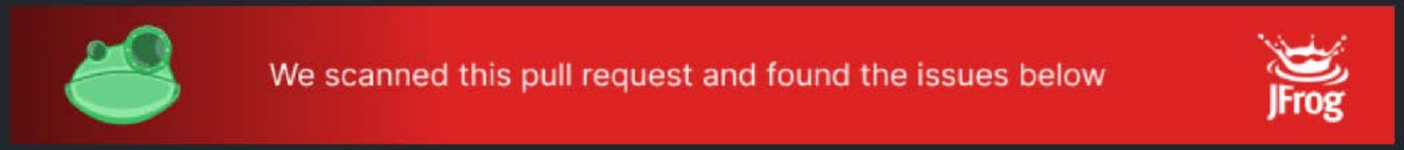
👍 No issues

If no new vulnerabilities are found, Frogbot automatically adds the following comment to the pull request:



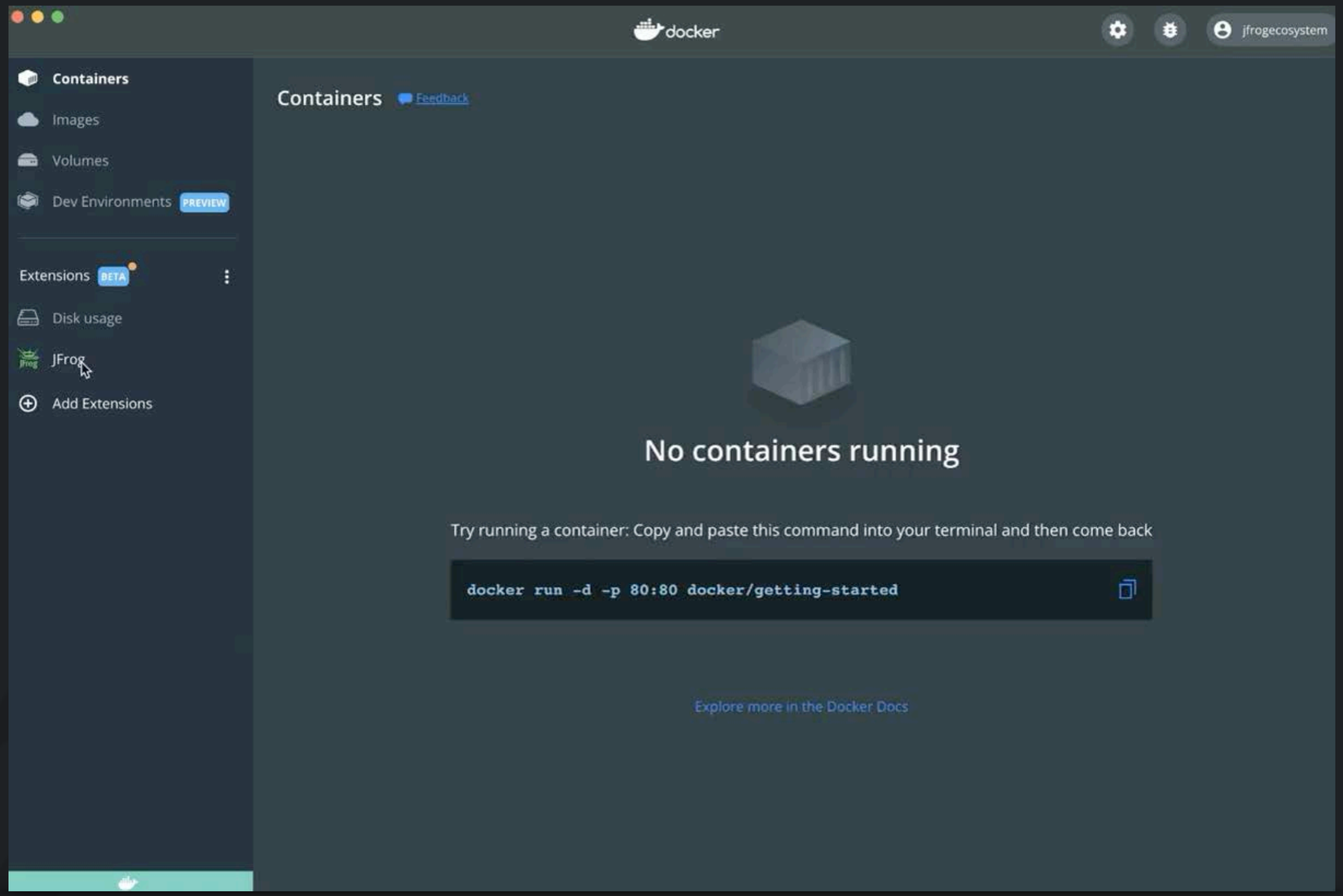
👎 Issues were found

If new vulnerabilities are found, Frogbot adds them as a comment on the pull request. For example:



SEVERITY	IMPACTED PACKAGE	VERSION	FIXED VERSIONS	COMPONENT	CON VE
High	github.com/nats-io/nats-streaming-server	v0.21.0	[0.24.1]	github.com/nats-io/nats-streaming-server	v
High	github.com/mholt/archiver/v3	v3.5.1		github.com/mholt/archiver/v3	
Medium	github.com/nats-io/nats-streaming-server	v0.21.0	[0.24.3]	github.com/nats-io/nats-streaming-server	v

https://github.com/jfrog/frogbot



https://jfrog.com/blog/get-peace-of-mind-about-security-when-deploying-containers-from-docker-desktop/

app – pom.xml

```
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>log4j</groupId>
            <artifactId>log4j</artifactId>
            <version>1.2.17</version>
        </dependency>
    </dependencies>
</dependencyManagement>
<dependencies>
    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
    </dependency>
</dependencies>
</project>
```

project › dependencyManagement › dependencies › dependency › artifactId

JFrog: Local CI

Dependencies (Issues #)

log4j:log4j:1.2.17 (6)

Impacted Component	Fixed Versions
log4j:log4j:1.2.17	[]
log4j:log4j:1.2.17	[]
log4j:log4j:1.2.17	[]
log4j:log4j:1.2.17	[]
log4j:log4j:1.2.17	[]
log4j:log4j:1.2.17	[]

More Info

Group:	log4j
Artifact:	log4j
Version:	1.2.17
Scopes:	Compile
Licenses:	Unknown
Top Issue Severity:	Critical

IntelliJ Idea Plugin

README.md

jfrog-npm-tools

A collection of tools to help audit your NPM dependencies for suspicious packages or continuously monitor dependencies for future security events.

The tools:

1. npm-secure-install - Validate dependencies are locked down to the exact versions before installation of global tools
2. package-checker - Python command line tool that checks a dependency string for what will actually be installed and whether it is suspicious
3. npm_issues_statistics - Analyzes github comments to find unusual activity that might correlate to compromised dependency

https://github.com/jfrog/jfrog-npm-tools

@Conf42: DevSecOps 2022

A journey of the thousand binaries

Ixchel Ruiz


Dad.. That kid said my artwork sucked!
Oh yeah... Well I'm going to teach him a lesson!
HOW TO GIVE GOOD FEEDBACK
① Be Positive
② Be Specific
③ Give a Next STEP.
Good use of color, but I think it needs more detail.
I'll definitely consider that. Thanks.
www.lunarbaboon.com