Securing Open Source

Adapting Secure Software Practices to Open Source Projects

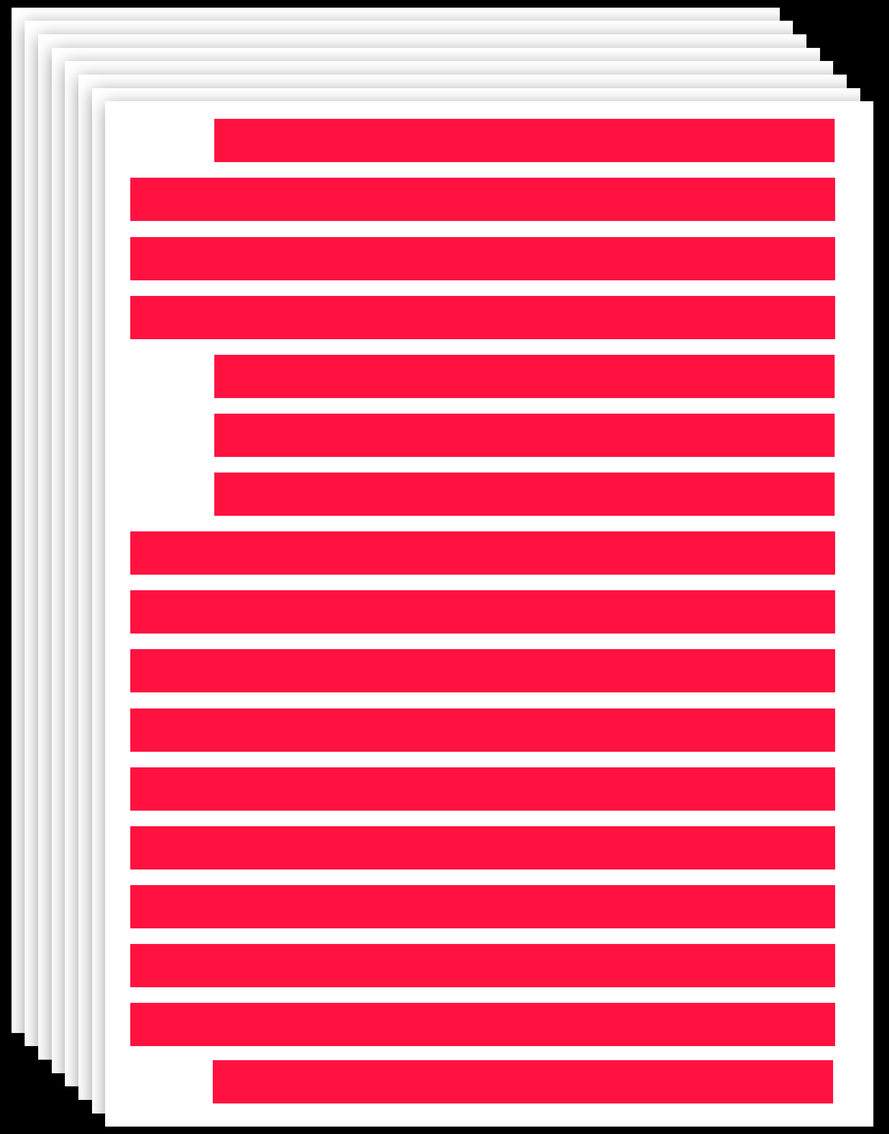
Alex Beaver
alexbeaver.com
/in/alex-beaver


LOG4J

POLICIES

STANDARDS

BEST PRACTICES



WHAT SHOULD FOSS MAINTAINERS TO DO TO SECURE PROJECTS?

Phases

How do you write a secure app?

What policies should FOSS projects use to guide security?

What tools can help maintainers with security?

1
FOSS vs Enterprise

2
Application Development

3
Release Management

DEVELOP SECURE SOFTWARE

DEVELOP SECURE SOFTWARE

SOFTWARE	TRAINING	PENTESTS
SDLC		
PSIRT		

FOSS maintainers have fewer resources and less control than enterprises

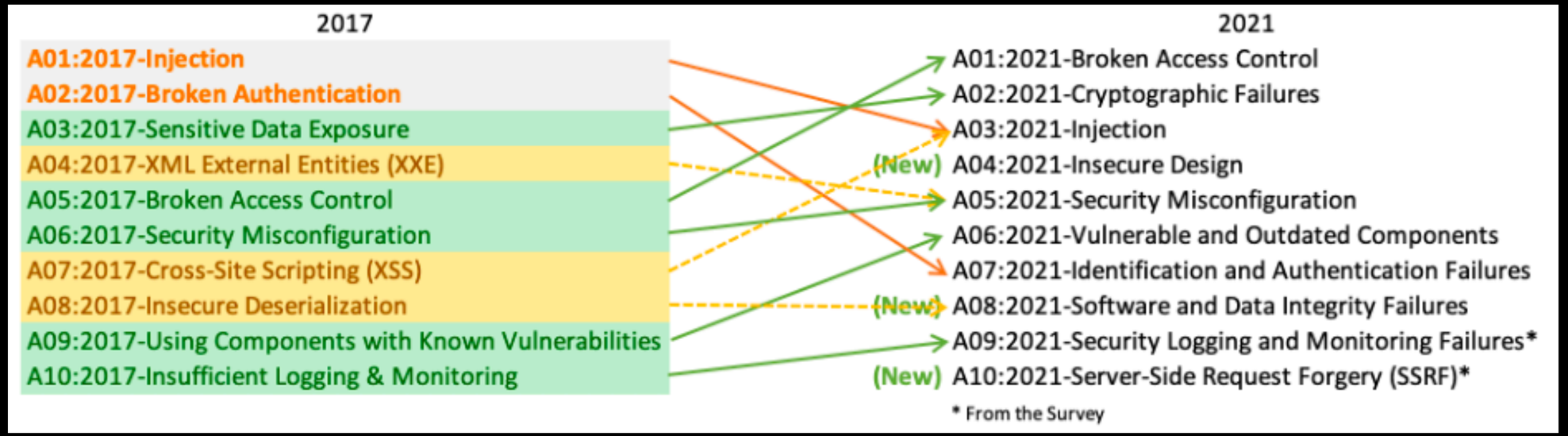
1
FOSS vs Enterprise

2
Application Development

3
Release Management

Software Design: Vital to Secure Software Hard to Verify

Baselines



beave.rs/owasp10

Provide Feedback

tensorflow

An Open Source Machine Learning Framework for Everyone

tensorflow | https://github.com/tensorflow/tensorflow | Version 2.15.0

Package information

- Visibility: **public**
- Forks count: **89,000**
- Stars count: **178,000**
- Default branch: **master**
- Open issues count: **2,050**
- Watchers count: **7,690**
- Contributors: **300**
- Blog: **http://www.tensorflow.org**
- Public Repos Count: **109**

Trusty Score

Repo Activity Score

Author Activity Score

How do we calculate this score?

We use statistical analysis to rate packages from 0-10. Higher ratings typically indicate safer packages.

Learn More

Provenance

The Python ecosystem does not support source of origin or build provenance verification, so Trusty is unable to verify that the package is related to its claimed repository.

Shared Repositories

Possible StarJack attack for package “tensorflow” The following packages list the same repository.

View full packages list →

Similarly Named Packages

All Clear. We were not able to locate packages with similar names.

tensorplot (Score 1.7) | **tensorflow1** (Score N/A) | **ytensorflow** (Score N/A) | **tensorflowjs** (Score N/A)

https://www.trustypkg.dev/pypi/tensorflow

Static AST

Looks at source code

Builds a model of the flow

Looks for known vulnerable patterns

High false positive rate

GitHub Code Scanning

Dynamic AST

Runs the application and looks for vulnerabilities

Fuzzing, Injection, etc.

Looks for likely vulnerabilities

May miss coverage

GitLab DAST, OWASP ZAP

1
FOSS vs Enterprise

2
Application Development

3
Release Management

Some of the greatest risks to FOSS security are at release

Track/Level	Requirements
Build L0	(none)
Build L1	Provenance showing how the package was built
Build L2	Signed provenance, generated by a hosted build platform
Build L3	Hardened build platform

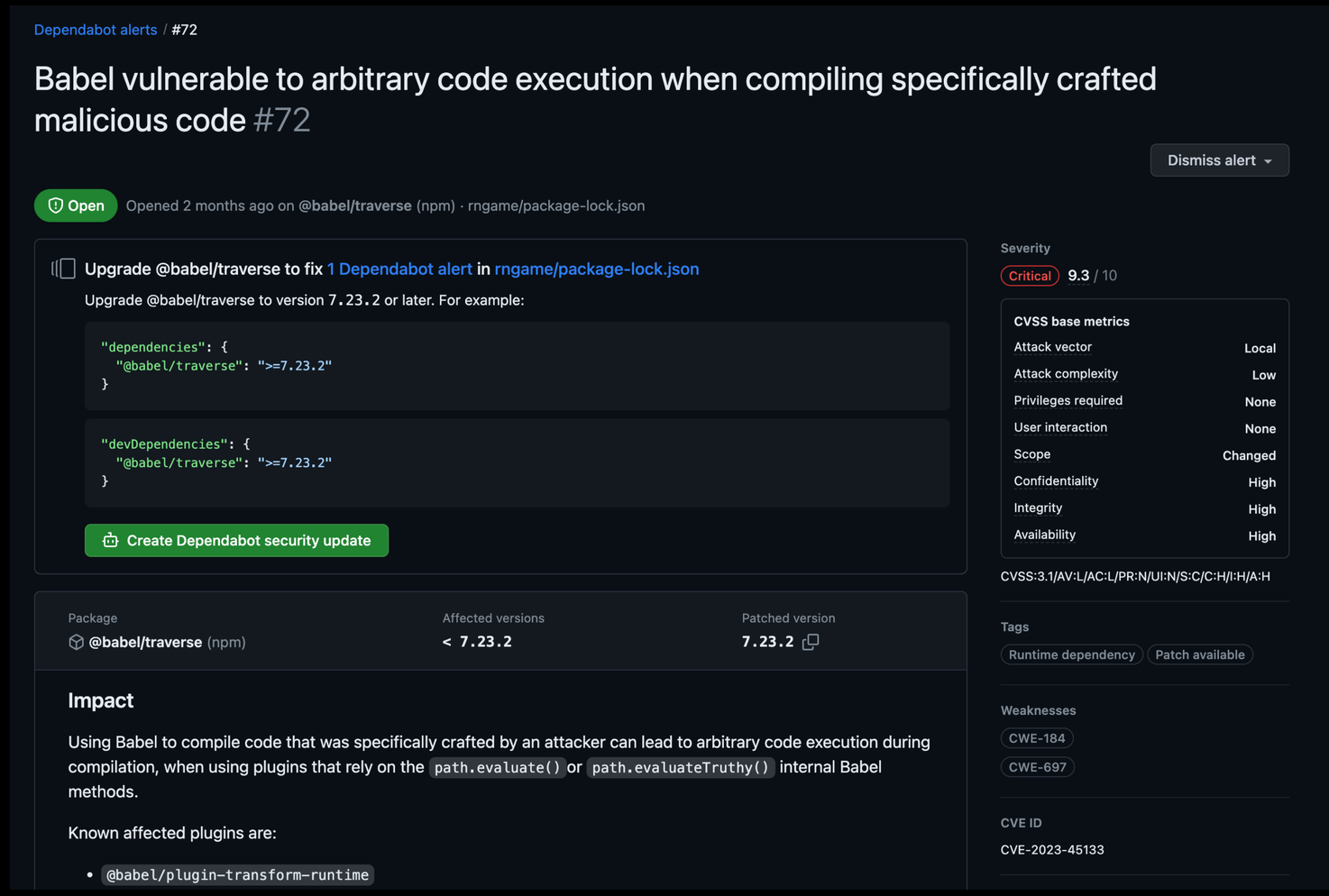
SLSA

Build L1: Provenance exists

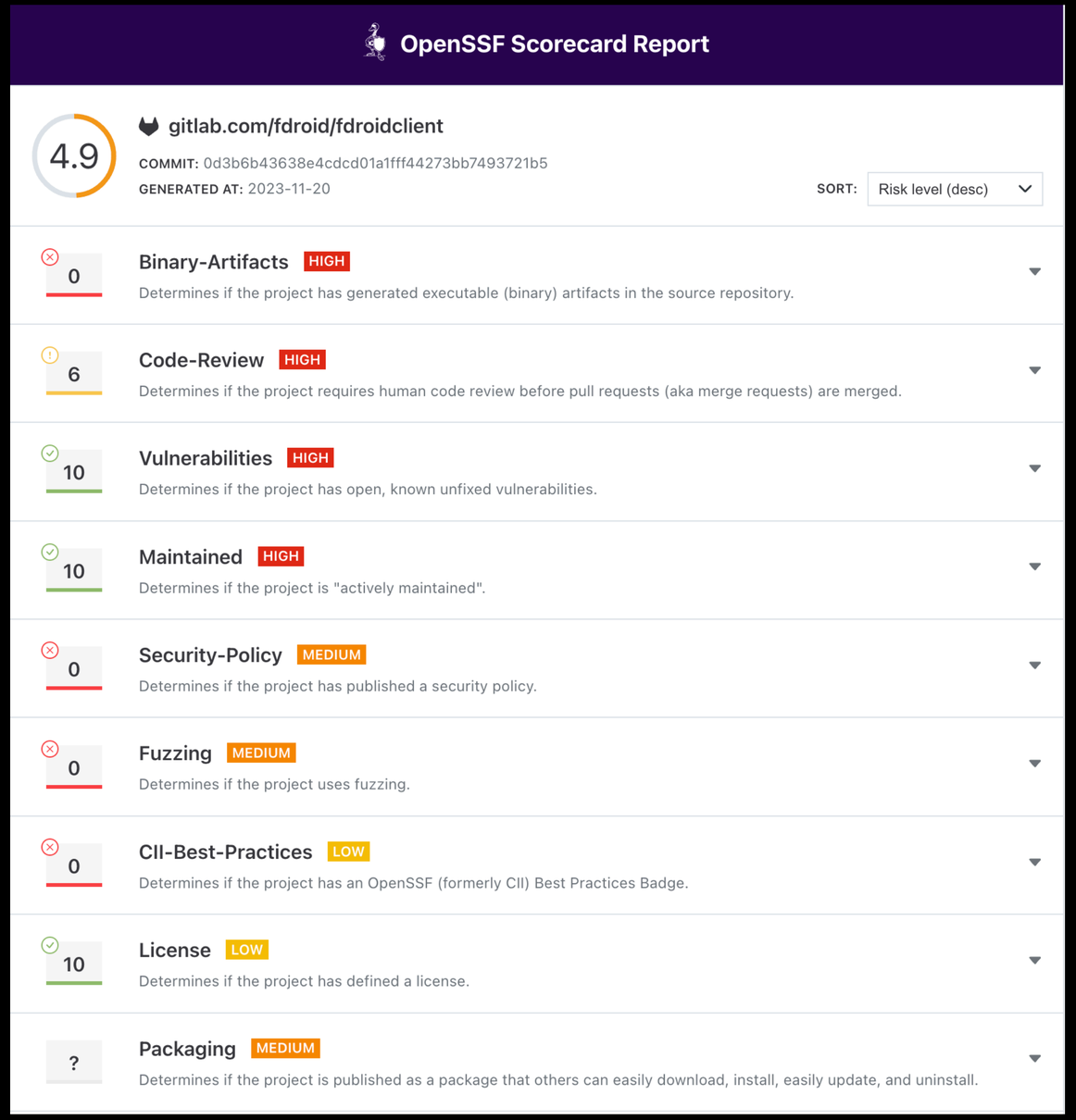
Summary	Package has provenance showing how it was built. Can be used to prevent mistakes but is trivial to bypass or forge.
Intended for	Projects and organizations wanting to easily and quickly gain some benefits of SLSA—other than tamper protection—without changing their build workflows.
Requirements	• Software producer follows a consistent build process so that others can form expectations about what a “correct” build looks like. • Provenance exists describing how the artifact was built, including the build platform, build process, and top-level inputs. • Software producer distributes provenance to consumers, preferably using a convention determined by the package ecosystem.
Benefits	• Makes it easier for both producers and consumers to debug, patch, rebuild, and/or analyze the software by knowing its precise source version and build process. • With verification, prevents mistakes during the release process, such as building from a commit that is not present in the upstream repo. • Aids organizations in creating an inventory of software and build platforms used across a variety of teams.
Notes	• Provenance may be incomplete and/or unsigned at L1. Higher levels require more complete and trustworthy provenance.

slsa.dev

Dependabot


Dependabot alerts / #72
Babel vulnerable to arbitrary code execution when compiling specifically crafted malicious code #72
Dismiss alert
Open
Opened 2 months ago on @babel/traverse (npm) · rngame/package-lock.json
Upgrade @babel/traverse to fix 1 Dependabot alert in rngame/package-lock.json
Upgrade @babel/traverse to version 7.23.2 or later. For example:
"dependencies": {
"@babel/traverse": ">=7.23.2"
}
"devDependencies": {
"@babel/traverse": ">=7.23.2"
}
Create Dependabot security update
Package
@babel/traverse (npm)
Affected versions
< 7.23.2
Patched version
7.23.2
Impact
Using Babel to compile code that was specifically crafted by an attacker can lead to arbitrary code execution during compilation, when using plugins that rely on the path.evaluate() or path.evaluateTruthy() internal Babel methods.
Known affected plugins are:
@babel/plugin-transform-runtime
Severity
Critical 9.3 / 10
CVSS base metrics
Attack vector Local
Attack complexity Low
Privileges required None
User interaction None
Scope Changed
Confidentiality High
Integrity High
Availability High
CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Tags
Runtime dependency
Patch available
Weaknesses
CWE-184
CWE-697
CVE ID
CVE-2023-45133

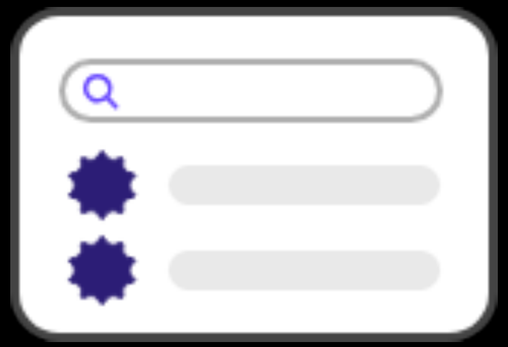
OpenSSF Scorecard



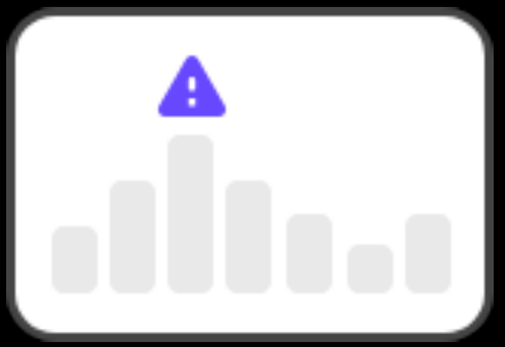
https://securityscorecards.dev/viewer/?uri=gitlab.com/fdroid/fdroidclient

Sigstore

Sign



Verify



Monitor

1
FOSS vs Enterprise

2
Application Development

3
Release Management

Next Steps

How is AI going to help or hurt secure development?

Are there open source alternatives to the proprietary software mentioned?

Is there any way that maintainers can help contributors with secure design?

alexbeaver.com

Connect With Me


/in/alex-beaver


@alexcbeaver