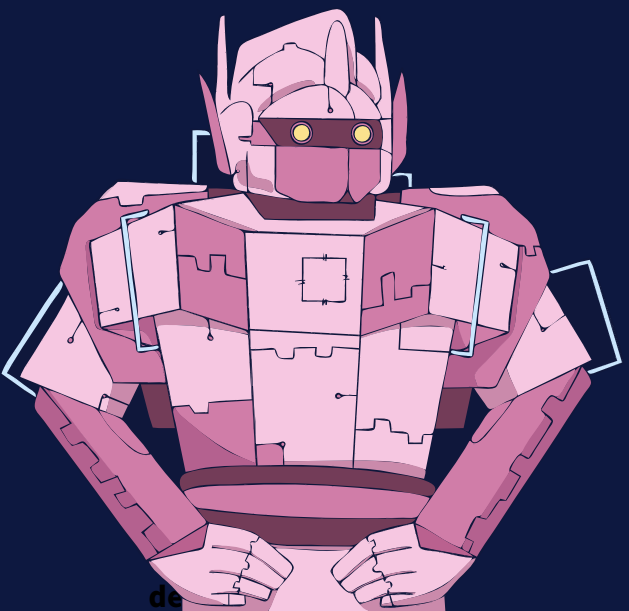
Managing and Fixing Open Source Vulnerabilities at Scale

Through Automation

Table of Contents



SOLVE vulnerabilities at scale

Dealing with open source vulnerabilities has become one of those boring, time consuming tasks that you sweep under the rug. If you don't look, it doesn't exist - right?

Not quite so, as while you are not looking, your security debt is building up - dependency by dependency. Keeping your dependencies secure and up to date is surely complex, but it has to be done. The question we all ask ourselves is how.

Is it even possible to do manually? Most experts would argue that no, it rarely is. When it comes to commercial software development, where the codebase usually is on the large side, it quickly gets out of hand. Oftentimes, the amount of vulnerability alerts are overwhelming and causes more damage than good.

This calls for tools, paired with the right processes, that will allow you to automate the way you handle security within your open source. In this ebook, we will walk you through a few important steps to help you manage open source vulnerabilities in an efficient manner, as

Debricked Team

WHY should I care about vulnerabilities in open source dependencies?

Today, all companies developing software use open source dependencies or other third party components to varying degrees. This adds a new dimension to software development since security issues in open source code will affect the products in similar ways as issues in the proprietary code. In the end, customers and hackers don't care what type of code was vulnerable.

Latest stats from

2022

500+

dependencies used in one open source project

96%

of all code bases relied on open source

76%

of a typical code base was open source

28 000+

new vulnerabilities disclosed in Debricked monitored sources

40% of which are related to OSS.
Monitored sources include NVD, GitHub Advisory database, and several language and package manager specific vulnerability datasets.

HOW can I manage security in my OSS efficiently?

In general, open source has the potential to be more secure than proprietary code, as all of the code is available publicly for inspection. The problem is that it assumes that someone is actually looking.

However, considering the amount of dependencies in popular frameworks being used in development, manual analysis is infeasible for most. So, the question still stands - how can one fix and prevent vulnerabilities from entering their codebase when it is impossible to manage manually?

SOLUTION: select the right tools to do it for you

Putting security in the hands of developers enables them to scan for vulnerabilities every time they push code. This in turn minimizes the risk of importing critical vulnerabilities through open source dependencies.

To do this, you need a Software Composition Analysis (SCA) tool that will assist you in detecting and patching vulnerabilities in the open source used in your applications.

A framework that will simplify the process

You also need some kind of framework that allows you to work with vulnerability management at scale. This is to streamline the procedure while you save time rather than lose. Not long ago, Google introduced a framework for tackling the challenge of vulnerabilities in open source dependencies, as well as industry wide goals for improving tooling capabilities.

The framework outlines three separate areas that you need to account for, namely:

KNOW the vulnerabilities in your software

PREVENT the addition of new vulnerabilities

FIX or remove vulnerabilities

The Debricked SCA tool makes applying the framework to your organization an easy task by delivering clear, actionable information to the right people at the right time. How we do that will be uncovered next.

KNOW

The “know” part of the framework is the core of any SCA tool helping you to manage vulnerabilities in open source. It entails mapping the dependencies used in your code toward one or many databases of vulnerabilities to find out if you’re using the vulnerable versions of the dependency.

In order to do this manually, you would constantly need to monitor all of the open source projects used in your application. More often than not, new vulnerabilities are found long after the initial import of dependencies. With or without changes to the projects, a one time scan is not sufficient.

Debricked helps you continuously monitor your projects and automaticlly alerts you when a vulnerability is discovered.

Don't drown in the stream of notifications

While getting notified is important, make sure that the tool you use gives you accurate information to limit the noise. One way of doing this is by using tools with high data quality and precision.

Debricked SCA is based on patented machine learning which enables both high data quality and coverage. As of today we offer over 90% precision in supported languages.

On top of this, we are able to automatically identify the vulnerable functionality used in your code, which eliminates false positives even further.

Debricked helps you:

- Automatically and continuously monitor your projects
- Stay updated about new vulnerabilities in your codebase
- Save time with over 90% precision in supported languages - almost no false positives!

PREVENT

For an open source consumer, “prevent” entails understanding risks when deciding on a new open source component.

In an organization, the decision-making process is often decentralized due to the lack of coordination among individual developers or relevant stakeholders. Risk reduction should therefore be approached from two directions:

- Managers should determine intolerable risks and establish policies to address the ones that already exist in the software pipelines.
- For developers, it should be easy to research risk before choosing a dependency to import.

Community health as a risk

The first, and most obvious, risk to consider is the one in the dependency version that you are considering using. You should also investigate the community, or “health”, of an open source dependency.

A healthy, active community has a greater chance at fixing vulnerabilities - and building safer open source projects to begin with. In contrast, a project that goes stale and has no one to find and fix vulnerabilities, will likely remain vulnerable longer.

A few things to consider when examining a community is the amount of contributors, experience of contributors, how fast they typically patch vulnerabilities and how mature their development practices are - do they maintain stable versions, for example?

In order to keep dependencies with poor health from being imported, you need to configure rules in a tool that blocks them. Debricked allows you to configure rules that fit your organization. You can choose to be notified, get a warning or even fail a pipeline when a rule is broken, to automatically keep your code base safe.

Debricked helps you:

- Set automated rules to prevent vulnerable dependencies from entering your codebase at CI/CD level
- Easily access information on the health of a dependency’s community using Debricked Select

FIX

There are a few ways to fix a vulnerability, with varying degrees of difficulty. You can develop a fix and contribute to the original project, fork the dependency and apply a patch, remove the dependency from your project or, ideally, update the dependency to a version that has already been fixed by someone else.

For most, updating the dependency to a safe version is the way to go and also where most tooling can help. With Debricked, you can do this with an automatic pull request.

Vulnerabilities in indirect dependencies

What complicates the matter is that, more often than not, you'll be vulnerable to an indirect dependency several layers down in the dependency tree.

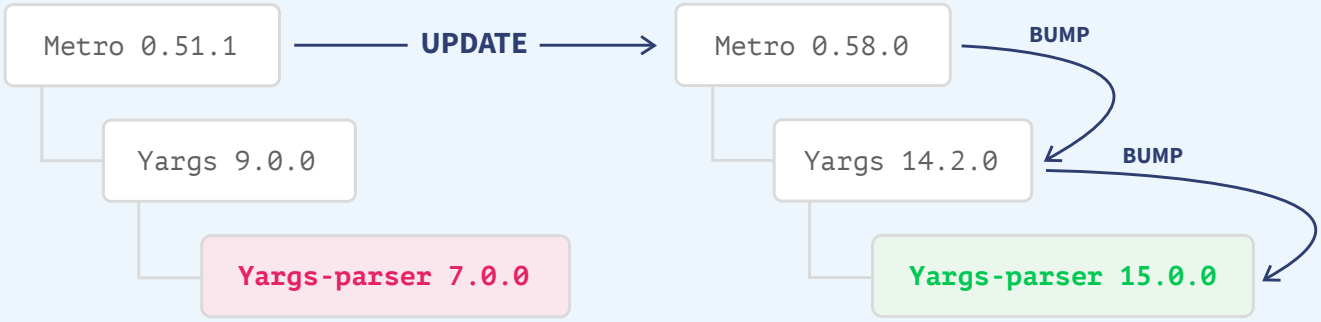
If you're indirectly using yargs-parser by importing it from a parent dependency, in this case Metro, knowing that version 15.0.0 fixes its vulnerability doesn't help you.

To solve it, you need to know which version of Metro that contains a version of yargs with the yargs-parser v.15.0.0.

Sounds tricky, huh?
Luckily, we're here to help.

Debricked helps you:

- Calculate root fixes that solve vulnerabilities in indirect dependencies far down in the dependency tree by updating the direct dependency
- Open Pull Requests for the fixes to remediate with just one click



AUTOMATION is the new black

Let's recap. So far we have outlined a framework that can help you manage open source vulnerabilities in a more efficient manner: Know, Prevent and Fix. But without the right tool and processes, the framework is null and void. This is where Debricked comes into the picture.

Today, the Debricked SCA tool can solve half of you vulnerabilities in just a few minutes and enable you to get license compliance control easily. Moving forward, we aspire to be the number one place where developers turn to get help choosing the best open source

Managing and fixing open source vulnerabilities has never been easier

We help companies all over the world achieve higher open source adoption while minimizing associated risks. We'd love to do the same for you.
Sign up for a free Debricked account and try it out. Looking for more information? Head over to our website to learn more.

Create your free account here

About Debricked

Debricked is redefining the way developers select, evaluate, use and contribute to open source software.
Debricked SCA and Select makes open source security, compliance and health simple, and helps companies use more open source in an efficient manner. Since March 2022, Debricked is a part of Micro Focus.

Radio Comms!

hello@debricked.com

Debricked AB, Minc,
Anckargripsgatan 3, Malmö