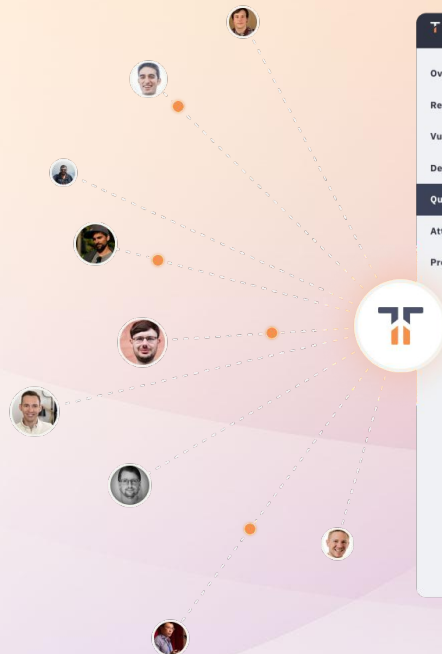


MAY 2024



## THE TIDELIFT GUIDE TO

# Reducing security risk from bad open source packages



The screenshot shows the Tidelift web application interface. At the top, there's a navigation bar with "TIDELIFT" and tabs for "Catalogs", "Projects", and "Packages". A left sidebar contains a menu with "Overview", "Releases", "Vulnerabilities", "Dependencies", "Quality Checks" (highlighted), "Attestation Data", and "Project Usage". The main content area is titled "Quality Checks" and includes a sub-header "Tidelift's research-backed checks help you understand the quality of an open source package". Below this is a "Security" section with a progress bar indicating "100% of checks passed". A list of checks follows, each with a right-pointing chevron and a "Passed" status in a green pill:

- > No known vulnerabilities on latest release **Passed**
- > Discoverable security policy **Passed**
- > 2FA enabled at source repository **Passed**
- > 2FA enabled for package manager **Passed**
- > Release managers are reviewed **Passed**
- > Security vulnerabilities have recommendations **Passed**
- > Package cryptographically signs releases **Passed**
- > Package uses fuzzing tools **Passed**

At the bottom, there is a section for "Development practices" with a crossed-out icon.

# What's covered in this guide



## **03** INTRODUCTION

**04.** The benefits of open source

**05.** The hidden cost of bad packages

## **06** HOW ARE MOST ORGANIZATIONS MANAGING OPEN SOURCE SOFTWARE RISK TODAY?

## **09** HOW CAN ORGANIZATIONS REDUCE SECURITY RISK FROM BAD OPEN SOURCE PACKAGES?

## **11** THE TIDELIFT MAINTAINER ADVANTAGE

## **13** CASE STORY: THE ROI FOR PROACTIVELY IMPROVING OPEN SOURCE SECURITY

## **14** FASTEST PATHS TO VALUE WITH TIDELIFT

## **19** MAINTAINER CASE STORIES

## **22** ABOUT TIDELIFT

## **24** GETTING STARTED



# Introduction

## INTRODUCTION

# The benefits of open source

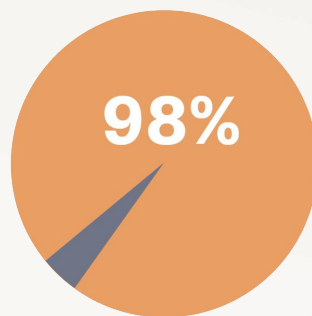
## Open source is the modern application development platform

These days, open source is everywhere.

Using open source gives anyone trying to innovate with software a head start, with billions of lines of code freely available, developed, and shared through an open community of creators, collaborators, and maintainers.

Open source helps increase developer productivity, accelerates development and deployment, and reduces application development costs.

However, it comes with hidden costs related to keeping it secure and well maintained.



of applications  
maintain open source  
components<sup>1</sup>



open source code  
makes up 70% or more  
of the average application<sup>2</sup>

<sup>1</sup> Sources: [Tidelift](#) and [Harvard](#)

<sup>2</sup> Sources: [Tidelift](#) and [Synopsis](#)

## INTRODUCTION

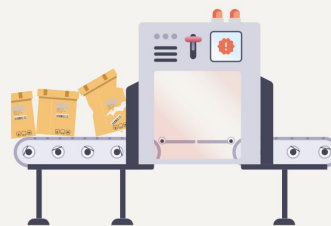
# The hidden cost of bad packages

Many orgs don't have a continuous view of where abandoned, insecure, or end-of-lifed packages exist in their applications.

These bad-for-enterprise-use packages create **security risk** that could potentially impact the organization's revenue, data, and business continuity.

They also **suck up valuable development cycles** when organizations have to replace them, work around them, or deal with endless cycles of vulnerability remediation.

### WHAT MAKES A PACKAGE "BAD" FOR ENTERPRISE USE?



We use the word *bad* as shorthand for a risky package that may lead to bad security outcomes or slow down development.

A package may be bad for enterprise use if it is unmaintained, deprecated or end-of-lifed, missing published security policies, unresponsive to security issues, or has been removed from the package manager.



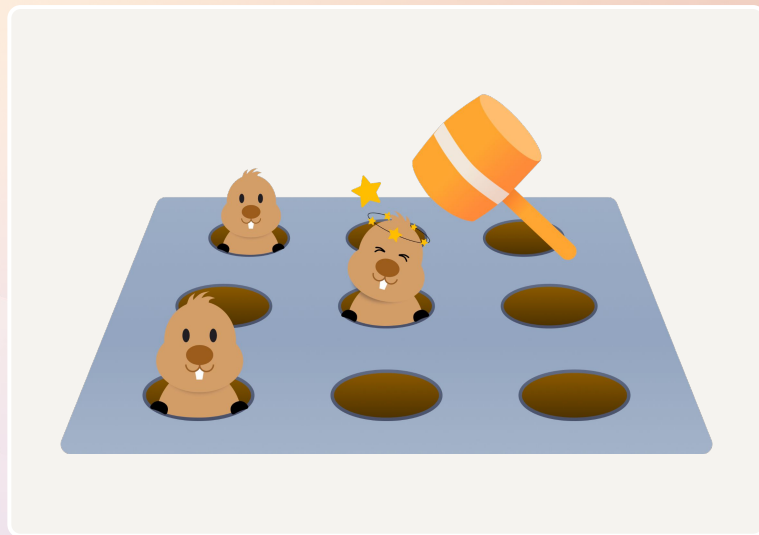
How are organizations  
managing open source  
software risk today?

# A reactive approach: vulnerability remediation

Many organizations use **software composition analysis (SCA) tools** to identify and remediate open source vulnerabilities.

*This is an effective way to ensure risk from existing vulnerabilities is reduced.*

**But it is also a game of whack-a-mole,** involving triaging long lists of security vulnerabilities that are difficult to prioritize and separating false positives from real security risks.

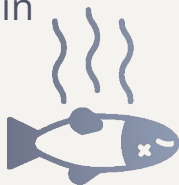


# Flagging known vulns: necessary, but not sufficient

SCA tools have been a common way for organizations to manage open source security issues, **but are only part of the solution.**

## DON'T EAT SPOILED FOOD

There are many options for helping your organization identify and fix known vulnerabilities in open source.



AND

## SOURCE BETTER QUALITY FOOD

But you also can make active decisions to bring in open source components that are being developed securely in the first place!







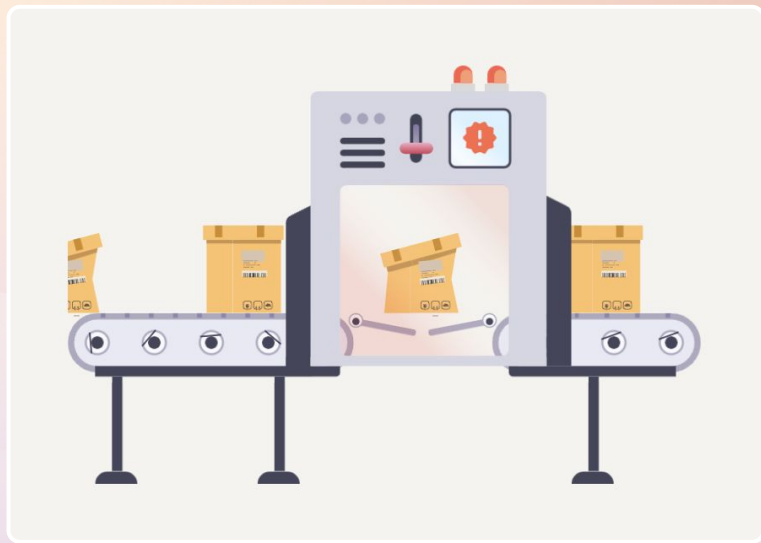
How can organizations reduce security risk from bad open source packages?

# A proactive approach: reducing reliance on bad packages

Tidelift takes a unique, data-driven approach to reducing reliance on bad packages:

We **partner with the maintainers** of thousands of the most-relied-upon open source packages and we **pay them** to implement industry-leading secure software development practices and document the practices they follow.

The result is a valuable source of cross-ecosystem package intelligence that customers can use to **identify and eliminate bad packages** AND **ensure the packages they rely on most keep getting better**.



# The Tidelift maintainer advantage

Tidelift is the **only** company that partners with open source maintainers and **pays them** to:

- Implement industry-leading secure software development practices and validate the practices they follow so organizations can have the same confidence in the security of their open source that they have in their own code.
- Contractually commit to continue these practices into the future so that organizations can confidently make long term investments in the packages they use.

**com.fasterxml.jackson.core:jackson-databind**  
General data-binding functionality for Jackson: works on core streaming API

This package was renamed. Previous names include: [maven/org.codehaus.jackson:jackson-mapper-asl](#), [maven/org.codehaus.jackson:jackson-mapper-lgpl](#), and [maven/com.codehaus.jackson:jackson-mapper-asl](#)

**This package is lifted!**  
Tidelift pays the maintainers of this package to uphold secure and sustainable development practices for the foreseeable future. You should feel confident that this package is enterprise-ready.

**Maintainer commitments**

- Vulnerability fixes for the latest release
- Secure vulnerability disclosure process
- Dependencies monitored for issues
- 2FA enabled on GitHub
- 2FA enabled in package manager
- Verified, machine readable license
- Defined security maintenance scope

**Releases** [View all](#)  
Most recent: **2.16.1** (23 Dec 2023)  
192 total releases since 2012.  
The last release was a month ago.  
[See full version guidance](#)

**Vulnerabilities** [View all](#)  
Latest: **2023-06-14** 4.7 Medium  
Most severe: **2019-01-02** 10.0 Critical

**Dependencies** [View all](#)  
Total dependencies: **9**  
Runtime: **2**  
Development: **0**  
Other: **7**

**Project Usage** [View all](#)  
Total projects using: **3**  
Direct use: **0**  
Transitive use: **3**

**License**  
Apache-2.0 (SPDX) [View details](#)  
Verified by maintainer

**Links**  
[Homepage](#) [GitHub](#) [Maven](#)

**Policies**  
[Security policy](#)

**Contributors (280)**  
  
+268 contributors [View all](#)

# Tidelift helps leading organizations use open source with confidence.



## **Reduce security risk**

by eliminating attack entry points through bad packages.



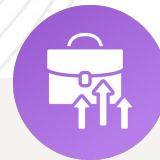
## **Improve productivity**

by reducing vulnerability fire drills from insecure or undermaintained packages.



## **Improve application quality**

by building with healthy and resilient open source packages.



## **Increase operational efficiency**

by saving costly manual package evaluation time.

# Case story: the ROI for proactively improving open source security

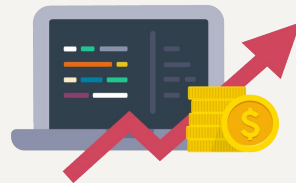
One large enterprise customer had a strategic initiative to **reduce time and money spent** managing the open source dependency lifecycle.

Scanning to discover and fix vulnerabilities was **not** getting them to the goal fast enough.

So they undertook an effort to **prioritize proactively preventing bad packages from entering production**, while also cleaning up risk from packages already in place.

## RETURN ON INVESTMENT

- Saved \$1,650,000 in time that would have been otherwise spent manually evaluating packages
- Avoided 50,000 bad releases that had been formally deprecated, abandoned, or were otherwise unfit for use
- Benefitted from 300 vulnerabilities fixed by maintainers in accordance with Tidelfit's maintainer contract
- The remediation of these vulnerabilities by partnered maintainers removed 3000 points of risk in applications running in production



# Fastest paths to value with Tidelift



## **Evaluating packages**

before pulling them in for  
application development



## **Actively monitoring**

the open source  
packages in use



## **Identifying and eliminating bad packages**

already adopted



## **Reinforcing at-risk packages** to keep them from becoming bad





# Evaluating packages before pulling them in for application development

When researching and evaluating open source packages to use, Tidelift's package recommendations provide an excellent starting point. The recommendation is a holistic evaluation of the package, and whether it is developed and maintained in a way that would make it a good fit for application development.

**It is also easy to undertake deeper package analysis with answers to questions such as:**

- ❓ Is it actively maintained or is it deprecated?
- ❓ Are the maintainers actively responding to security issues?
- ❓ Does it conform to my organization's license policies?

```
"description": "A querystring parser that supports nesting and  
arrays, with a depth limit",  
"tidelift_recommendation": "recommended",  
"versioning_scheme": "semver",  
"security_policy_url": "https://tidelift.com/docs/security",  
"contributors_count": 62,  
"package_manager_url": "https://www.npmjs.com/package/qs",  
"sdlc_policy": "https://support.tidelift.com/hc/en-  
us/articles/4406288074260-Lifter-tasks-overview",  
"sdlc_evidence": "https://support.tidelift.com/hc/en-  
us/articles/4406288074260-Lifter-tasks-overview",  
"repository": {  
  "url": "https://github.com/ljharb/qs",  
  "source": "package_manager"  
},  
"license": {  
  "expression": "BSD-3-Clause",  
  "source": "human_verified"  
},  
"latest_release": {  
  "version": "3.0.0",  
  "published": "2020-08-17T14:12:18.000Z"  
},  
"latest_stable": {  
  "version": "3.0.0",  
  "published": "2020-08-17T14:12:18.000Z"  
}
```

SECURITY 100%		DEVELOPMENT 100%		A LONG-TERM OUTLOOK 100%	
Package is not deprecated		Package is not deprecated		About this check	
Package appears maintained		Package has a stable release greater than two years old		This check indicates if the package has been marked as deprecated.	
Responsive to security issues		Why this check matters		Deprecated packages are unlikely to receive updates if a vulnerability or other issue is identified.	
		Result		Status: Passed	



# Actively monitoring the open source packages in use

Open source packages are constantly changing and it is important to monitor and review updates after making the initial decision to use a package. **Tidelift makes it possible to identify bad packages through early warning signs such as:**



New release availability, leading to end-of-support for older versions



New versions released under different license types



Package maintenance status changes



Packages or versions getting impacted by vulnerabilities

The screenshot shows the npm page for the 'angular' package. A magnifying glass icon highlights a warning sign. The page includes sections for Overview, Releases, Vulnerabilities, Dependencies, Quality checks, and Attestation data. A prominent pink box states: "This package is deprecated. NPM page says: 'This package has been deprecated. Author message: For the actively supported Angular, see https://www.npmjs.com/package/angular'. Github repo is archived and README says: 'AngularJS support has been deprecated.'". Below this, a vulnerability alert for CVE-2023-35116 (4.7 Medium) is shown. The alert includes a remediation note: "The maintainer of this package has marked this vulnerability as a false positive. The unaffected versions of this package are >= 2.15.3." and insights from the maintainer: "Provided for Tidelift subscribers on 11 Jul 2023 by the maintainer of com.fasterxml.jackson.core:jackson-databind." The alert also includes a link to the NIST information: "CVE-2023-35116".

Remediation	NVD published date
The maintainer of this package has marked this vulnerability as a false positive. The unaffected versions of this package are >= 2.15.3.	14 Jun 2023

Insights from the maintainer	NVD last modified
Provided for Tidelift subscribers on 11 Jul 2023 by the maintainer of com.fasterxml.jackson.core:jackson-databind.	7 Dec 2023 at 7:23 pm

Is this a real vulnerability or a false positive?	NIST information
This is a false positive. As per explanation on issue reported filed (https://github.com/FasterXML/jackson-databind/issues/3972) there is no actual reproducible vulnerability: code as shown is not something attacker can use; submitter basically shows how user can DoS their own service by trying to serialize a cyclic data structure specifically constructed as such. There is nothing to show how external caller could achieve that.	CVE-2023-35116





# Identifying and eliminating bad packages already adopted



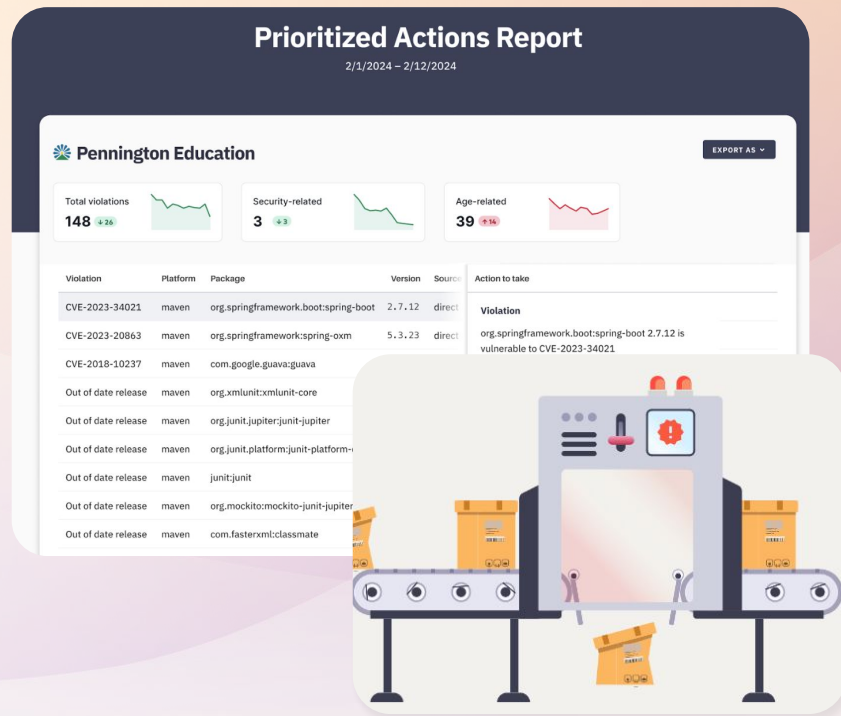
While it is ideal to identify and avoid bad packages in the first place, most organizations will have already adopted a significant number of packages without having done the upfront research.



Tidelift helps organizations evaluate their existing open source dependencies and prioritize the work to migrate away from bad packages.



Many maintainer partners also provide additional insights that can be used for prioritizing vulnerability remediation





# Reinforcing at-risk packages to keep them from becoming bad





# Maintainer case stories



## Case story: minimalist

Maintainer Jordan Harband saved minimalist from deletion when its maintainer decided to erase their projects from GitHub.

**minimalist** is a highly relied upon JavaScript package used by many Tidelift customers and was a transitive dependency the Tidelift maintainer partner **Jordan Harband** pulled into his projects.

The minimalist maintainer did not want to comply with npm's new 2FA requirement because he did not own a mobile phone, and rather than comply, **he decided to simply delete his entire GitHub account**, which would have impacted any organization that relied on minimalist or his other packages.

Jordan stepped in to take over ownership of minimalist and ensure it meets enterprise-grade secure software development practices. **Thanks to income from Tidelift and its customers**, Jordan has been able to ensure minimalist stays secure and reliable, and he was even able to bring on a co-maintainer for the project.



## Case story: jackson-databind

Maintainer Tatu Saloranta used income from Tidelift and its customers to completely re-architect jackson-databind and eliminate the risk of RCE vulnerabilities.

**jackson-databind** is an extremely popular general purpose java data-binding package that is important to many Tidelift customers, several of whom had expressed concerns that it was being impacted by a large number of remote code execution (RCE) vulnerabilities. At least one customer was planning to re-architect jackson-databind out of their infrastructure because of this increased risk.

**Thanks to income from Tidelift and its customers**, Tatu was not only able to implement enterprise-grade secure software development practices for jackson-databind, he was able to re-architect to eliminate the risk of RCE vulnerabilities once at for all.

The customer previously planning to re-architect reported that the risk score had been reduced so much **they no longer needed to replace the package**.



# About Tidelift



# About Tidelift

Tidelift helps organizations reduce risk to their revenue, data, and business continuity by proactively improving the resilience of the open source powering their applications.

Tidelift partners with leading open source maintainers and pays them to implement industry-leading secure development practices and validate the practices they follow. With Tidelift, organizations have the tools and intelligence they need to confidently make long-term investments in the open source they depend on.

## SELECTED CUSTOMERS



DARPA



## INVESTORS



# Getting started



## Watch a demo of the Tidelift Subscription

See how the Tidelift Subscription can help your organization avoid bad packages and use open source with confidence, so you can create more incredible software, even faster.

[WATCH THE DEMO](#)



## Learn about the Tidelift maintainer advantage

Read maintainer case stories and learn how paying the maintainers *works*.

[READ THE STORIES](#)

Get detailed technical information about the Tidelift Subscription.

[VISIT OUR TECHNICAL DOCUMENTATION](#)



## Get in touch

Contact us to schedule a time to chat live and learn more about how Tidelift can help you.

[CONTACT US](#)