# TIDELIFT
# 2023 OPEN SOURCE MAINTAINER IMPACT REPORT

**June 2023**

Tidelift's first annual overview of the strategy and key results from securing the open source software supply chain in partnership with open source maintainers.

TIDELIFT

# Contents

Over the past several years, Tidelift has built a model for partnering with open source maintainers to achieve improved software supply chain security outcomes across critical open source software ecosystems. In this report, we walk through the strategy, key outcomes to date, and highlight the opportunities and challenges ahead.

INTRODUCTION

# Aligning incentives with secure outcomes for open source

As noted in the U.S. National Cybersecurity Strategy, released in March 2023, cybersecurity is essential to our economy, critical infrastructure, data and communications privacy, as well as national defense. Recent software supply chain compromises highlight critical risks and the far-reaching impact of cybersecurity vulnerabilities to industry, government, and end consumers that must be addressed.

Against this backdrop, the U.S. government and other government and industry leaders around the world are taking action. In the U.S., a series of executive actions have resulted in a government-recommended secure software development framework that all software producers doing business with the government will need to attest that they follow. Other industry leaders like OpenSSF have also codified a set of security standards that apply specifically to open source projects.

**One thing that is clear from all of this momentum and activity is that there is broad consensus that urgent action is needed to ensure secure development practices in open source continue to improve.**

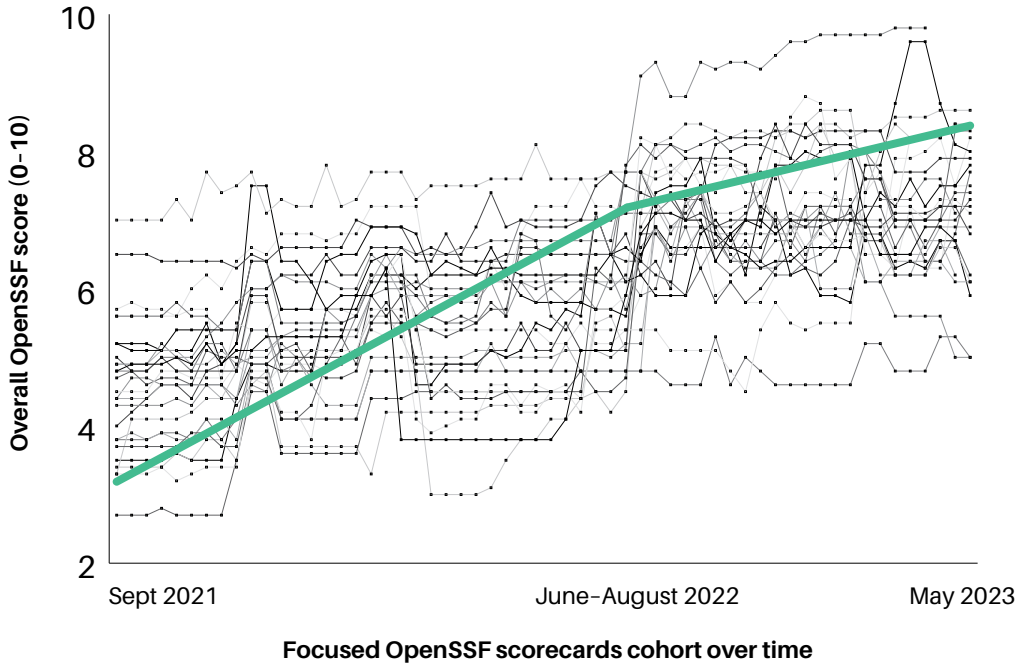
The NIST Secure Software Development Framework and recent reports from The National Security Telecommunications Advisory Committee (NSTAC) show the degree to which industry must understand, and attest to, the open source components that are going into their commercial software. NIST is clear that "secure by design" comes from ongoing intent and process across the organizational roles, responsibilities, and systems that go into our software, and NSTAC concludes that while open source software is not inherently less secure than commercial software, "incentives to invest in securing open source are neither effective nor sufficient."

Over the past several years, Tidelift has championed a model that pays open source maintainers directly to ensure their projects follow secure development practices like those codified in the NIST SSDF and the OpenSSF Scorecards project.

In this report, we will share our strategy, learnings, and results we've taken away from creating reliable incentives for maintainers to hold packages to a set of secure development standards.

Some key highlights in our results data include:

## Average OpenSSF secure development practices score improvement 57% since Fall 2021




**Focused OpenSSF scorecards cohort over time**

## Overall performance of Tidelift partnered maintainers on key security practices

100% of known vulnerabilities have had a fixed release made and/or documented mitigations to address known risk

96% have enabled two-factor authentication (2FA)

Tidelift partnered maintainers have a **37% higher rate of having security policies and resulting procedures in place** as compared to other OpenSSF scorecards-assessed packages.

THE CHALLENGE

# Incentives to invest in improved open source security outcome are not yet sufficient

## Government-private sector collaboration will be critical to accelerate progress

The National Security Telecommunications Advisory Committee's (NSTAC) November 2021 report to the President, *Software Assurance in the Information and Communications Technology and Services Supply Chain*, identified key challenges, opportunities, and recommendations across three main areas of focus: software assurance, stakeholders, and external influencing factors. Highlighted below are key findings and recommendations for open source software:

### Software assurance

*Table 1: Software assurance findings (portion)*

| Finding | |
|---|---|
| Open source software (OSS) is not inherently less secure than closed source software, but incentives to invest in securing open source are neither effective nor sufficient. | Open source software, which provides components for virtually all software products, thrives on diversity of contributions and contributor motivations. Not all contributors are motivated to adopt security assurance practices.<br><br>Developers and administrators may not have insight into the level of security assurance for OSS modules.<br><br>Various promising efforts are underway that may lead to improved trustworthiness and increased confidence for integrators and users of software products that contain open source. The prospects for success and impact of these efforts are still uncertain. |

## Software assurance (portion)

*Table 2: Software assurance recommendations (portion)*

| Recommendations | |
|---|---|
| 1.4. Improve security and assurance processes for OSS. | Incentivize collaboration between open source developers and organizations focusing on security, such as the Open Source Security Foundation (OpenSSF).<br><br>Task NIST to extend efforts from its work related to EO 14028 to identify the open source packages used for "critical software."<br><br>Task the Federal Government to engage with organizations, allied nations, and Government agencies outside of the U.S. (e.g., the European Union Agency for Cybersecurity [ENISA], the G7, or the United Nations), to create and fund a public-private software assurance program to improve open source security.<br><br>Develop standards to accurately describe software components, in collaboration with organizations such as OpenSSF and international standards bodies.<br><br>Encourage developers to adopt a system of code vetting, such as OpenSSF's Scorecard 2.0. |

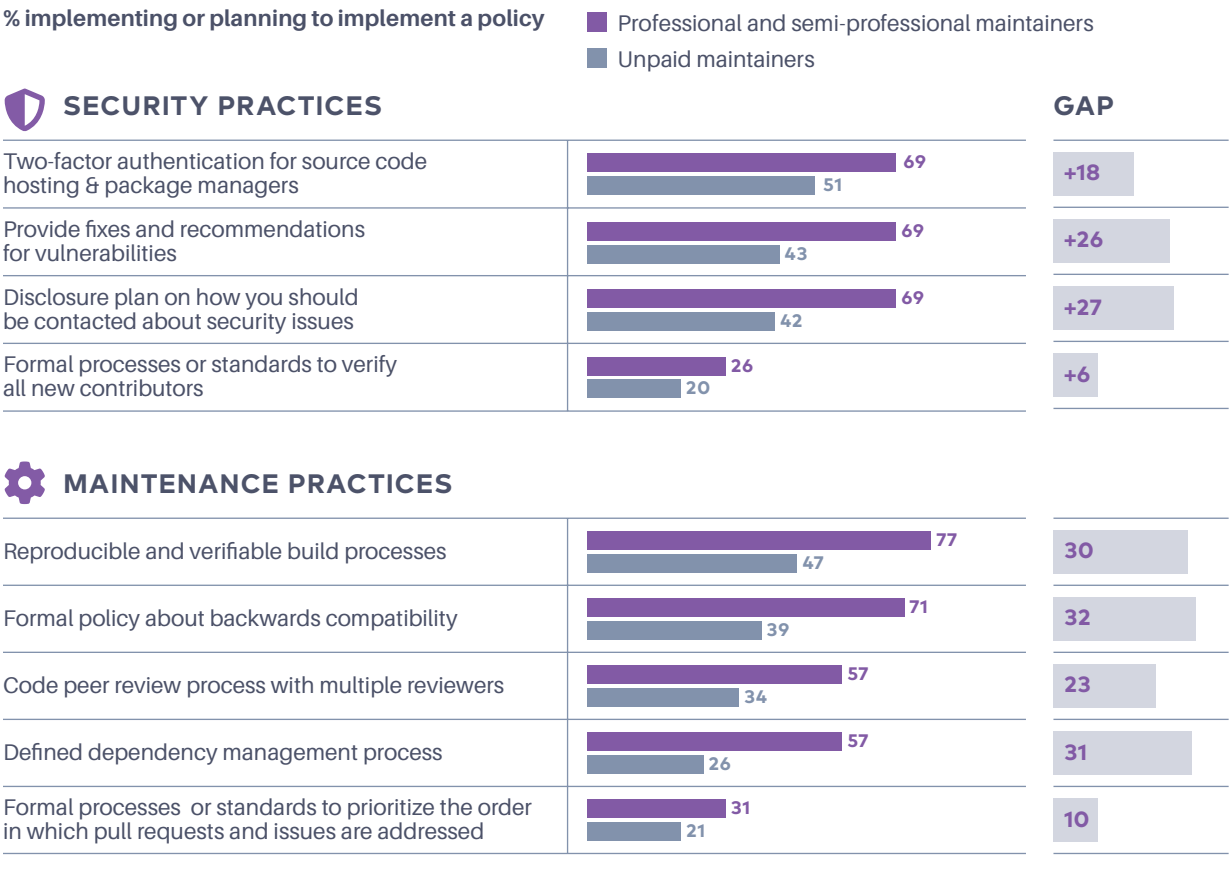
Source: https://www.cisa.gov/sites/default/files/publications/NSTAC%20Report%20to%20the%20President%20on%20Software%20Assurance.pdf

Tidelift's *2023 state of the open source maintainer report* shows that the gaps between unpaid and paid maintainers on important security and maintenance practices are significant. The charts below show the distance between work that is both actively happening, and planned work — paid maintainers were **25-30% more likely to have completed the work or plan to compared to unpaid maintainers**. On every development practice surveyed, paid maintainers were significantly more likely to be implementing it than unpaid maintainers.

We cannot secure our software supply chain without increasing the adoption of these practices, and others outlined in the NIST SSDF, due to the networked nature of open source software. A single package that doesn't have the time or incentives to undertake these practices can have enormous blast radius within modern software.

## Paid maintainers complete security and maintenance tasks more often than unpaid maintainers

Which of the following practices have been implemented for most or all of the projects that you maintain? Of the practices that have not been implemented, which would you consider implementing in the future?



SOLUTIONS

# Tidelift's model with maintainers has proven that secure-by-design must include relationships and incentives

## A combination of software + people will achieve the best results in securing the open source software supply chain

Tidelift's model starts with a set of clear standards for maintainers to uphold on a per-package and per-release basis. These standards are a best-in-industry set of development practices grounded in the Center for Internet Security Software Supply Chain Security Guide, OpenSSF Scorecard, and NIST Secure Software Development Framework.

When a maintainer, or maintenance team, signs a contractual agreement with Tidelift to uphold these standards, we provide an assessment of how a package is measuring up to the standards and steps to move toward compliance.

Over time, we see that our standards are improving objective measures of security, such as those provided by the OpenSSF Scorecard. We're also consistently working *with* maintainers to ensure that we're measuring what matters — practices that actually improve security outcomes.

The following pages describe the standards and impact at the core of the Tidelift model. Tidelift is using this list of standards to provide guidance to partnered maintainers in the way any software producer would guide the compliance and development practices of their in-house development teams. This model has shown that — across ecosystems, and at any project size — a central set of standards can be applied and upheld by maintainers, when the right incentives are in place.

The scale of this model today includes over 5,000 commonly used packages across Java, Python, JavaScript, PHP, Nuget, Rust, Ruby, and other ecosystems. These packages comprise nearly 50% of our paying customers' supply chains today.

Below is the Tidelift framework showing what makes an open source package secure and resilient, grounded in industry standards and best practices:

| Standard | Whole supply chain outcome | Industry mapping |
|---|---|---|
| Maintainer identity, compliance with our code of conduct is verified | Package has been marked as supported | NIST SSDF PO.1.1 |
| Business contract to ensure compliance with secure development practices | Package has guaranteed compliance with secure software development practices | NIST SSDF PO.1.1, PO.2.1; CIS Software Supply Chain Security Guide 3.2.4: Ensure packages are automatically scanned for ownership change |
| Package has discoverable security policy | Package has a security process to handle vulnerabilities | NIST SSDF PW.4.4, NIST SSDF RV.1.3; OpenSSF Scorecard check: Security-policy; OpenSSF Best Practices: Vulnerability Report Process; CIS Software Supply Chain Security Guide 1.2.1: Ensure public repositories have a SECURITY.md |
| Maintainers use 2FA to access the source code repository | Two-factor authentication greatly lowers the risk of account compromise, where a compromise can lead to trojaned or hijacked code. Protects supply chains from a breach, and subsequent trojan/hijacked code | NIST SSDF PW.4.4; CIS Software Supply Chain Security Guide 1.1.1: Ensure changes are tracked in a version control platform |
| Maintainers use 2FA to push releases to the package manager | Protects supply chains from a breach, and subsequent trojan/hijacked code | NIST SSDF PW.4.4; CIS Software Supply Chain Security Guide 4.2.3: Ensure user access to the package registry requires two factor authentication |
| Package has verified upstream repository source url | Provides a reference and record of changes in their package and dependencies | NIST SSDF PW.4.4; CIS Software Supply Chain Security Guide 4.2.3: Ensure user access to the package registry requires two factor authentication |
| Package has correctly defined package versioning scheme | Provides the necessary information to apply upgrades and mitigations | OpenSSF Best Practices: Versioning |
| Package has verified release manager role access | Protects package from trojan/hijacked code in their supply chain | NIST SSDF PO.1.3; CIS Software Supply Chain Security Guide 3.1.5: Ensure trusted package managers and repositories are defined and prioritized; CIS Software Supply Chain Security Guide 3.2.4: Ensure packages are automatically scanned for ownership change |
| Package has mapped security policies across release streams | Provides the necessary information to apply upgrades and mitigations | NIST SSDF RV.1.3; OpenSSF Best Practices: Versioning |
| Package has verified SPDX formatted, open source license | Accurate licensing data allows organizations and legal teams to apply automated compliance policies accurately, and at scale. | OpenSSF Scorecard check: License; OpenSSF Best Practices: License; CIS Software Supply Chain Security Guide 1.5.6: Ensure open source licenses are tracked |

| Standard | Whole supply chain outcome | Industry mapping |
|---|---|---|
| All vulnerabilities affecting a package have been identified | Focuses effort for packages that have identified vulnerabilities | NIST SSDF RV1.1, RV.1.3; CIS Software Supply Chain Security Guide 1.5.4: Ensure scanner in place for open source vulnerabilities; OpenSSF Scorecard check: Vulnerabilities |
| All releases affected by vulnerabilities have been verified and have a response | Provides a solution for deeply nested supply chain vulnerabilities, ensuring they have a solution for any existing vulnerabilities packages, and package dependencies, in the supply chain. | NIST SSDF PO.4.1; NIST SSDF RV.1.3, RV.2.1, RV.2.2; CIS Software Supply Chain Security Guide 1.5.4: Ensure scanner in place for open source vulnerabilities |
| Package is actively maintained | Package has maintenance team that is paying attention for potential bugs or security risks that could compromise the supply chain | NIST SSDF PO.4.1, PO.4.2; NIST SSDF PW.4.1, PW.4.2, PW.4.4; CIS Software Supply Chain Security Guide 3.1.4: Ensure dependencies are monitored between open-source components |
| No binary artifacts in source repository | Protects supply chains from potential maliciously subverted executables | OpenSSF Scorecard check: Binary-Artifacts |
| Packages have a safe release available, including safe dependencies | Gives end users the possibility of an upgrade path out of risk | NIST SSDF PW.4.4; CIS Software Supply Chain Security Guide 3.1.4: Ensure dependencies are monitored between open-source components |
| Package dependencies are clean | Takes care of direct and transitive risk down the supply chain of a package | NIST SSDF PW.4.4; CIS Software Supply Chain Security Guide 3.1.2: Ensure SBOM is required from all third-party suppliers; CIS Software Supply Chain Security Guide 3.1.4: Ensure dependencies are monitored between open-source components |
| Package has known income streams | Creates more assurance that the maintenance team has the resources needed to handle security responses and future development | NIST SSDF PW.4.4 |
| (future) Package has dependency management practice | Consistent high dependency update velocity is strongly correlated with improved project security | NIST SSDF PO.3.3, PO.4.1; OpenSSF Scorecard check: Dependency-Update-Tool |
| (future) Package has low Mean Time to Update Dependencies (MTTU) | Consistent high dependency update velocity is strongly correlated with improved project security | NIST SSDF PO.3.3, PO.4.1; OpenSSF Scorecard check: Dependency-Update-Tool |
| (future) Package has code review practices | Mitigate malicious or vulnerable code | NIST SSDF PW.4.2, PW.4.4; OpenSSF Scorecard check: Code-Review |
| (future) Package has a fuzzing practice | Regular, proactive checks for vulnerability or other injection weakness | NIST SSDF PW.4.2, PW.4.4; OpenSSF Scorecard check: Fuzzing |
| (future) Package has maintainer stability signified by recent activity on other maintained packages | Increased predictability on long term outlook | NIST SSDF PW.4.4 |

RESULTS

# Tidelift's model and outcomes deliver the assurances our software supply chain requires

These results are in alignment with both the NIST SSDF and the NSTAC findings and recommendations for software development practices, and open source software assurances

## Tidelift has driven maintainer adoption of a system of code vetting

The OpenSSF Scorecard was created to help open source maintainers improve their security best practices and to help open source consumers assess whether the packages they are using in their software are safe.

The scorecard is an automated tool that assesses a number of important heuristics ("checks") associated with software security and assigns each check a score of 0-10, as well as an overall 0–10 score. The team behind the scorecard runs a **regular analysis against millions of the most critical open source projects and publishes the resulting scores** in a BigQuery public dataset.

The limitations of the scorecard data include:

- today, the scorecard primarily scans project practices for projects hosted on GitHub, which doesn't include projects that host source code on other repositories such as GitLab, Sourceforge, etc for the majority of the checks
- the scorecard can only scan for activity that can be automatically detected, such as use of a particular dependency management tool ("Dependabot")
- this also precludes reporting on hidden—but no less important—activity such as enabling two-factor authentication (2FA), having succession plans in place for a package, or going deeper into how packages are built and shipped out into the software supply chain.

While the scorecard is starting to see awareness and adoption grow since 2021, when we surveyed open source maintainers in 2023, and review score data, **we aren't seeing broad enough awareness and adoption to secure the open source software supply chain:**

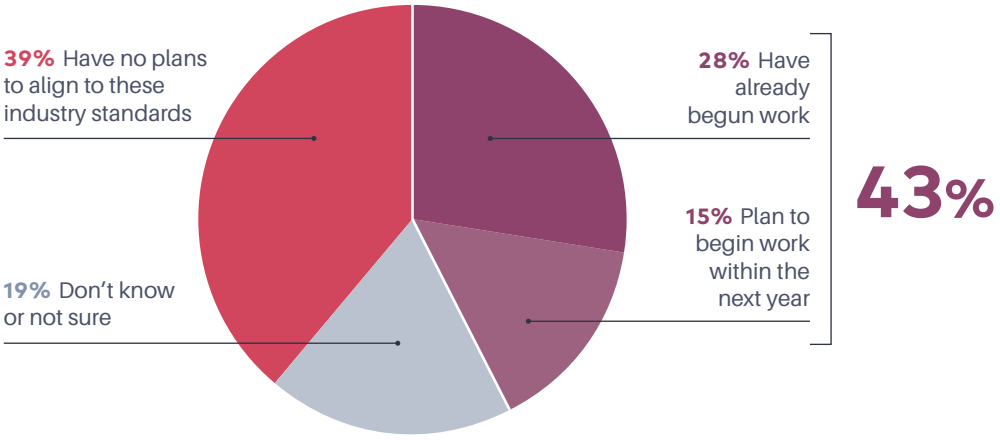
## Over half of maintainers are not aware of prominent software security standards

Which of the following industry standards initiatives are you aware of? (Choose all that apply)

| | |
|---|---|
| OpenSSF Security Scorecards | 28% |
| NIST Secure Software Development Framework | 26% |
| Supply Chain Levels for Software Artifacts Framework | 13% |
| None | 52% |

## 43% of maintainers aware of industry security standards have already begun or plan to begin work to align to one or more of them

Have you already or do you plan on beginning the work needed to ensure your projects align with one or more of these industry standards?




**3.3** Average score out of 10 in May 2023 for **all** evaluated open source packages

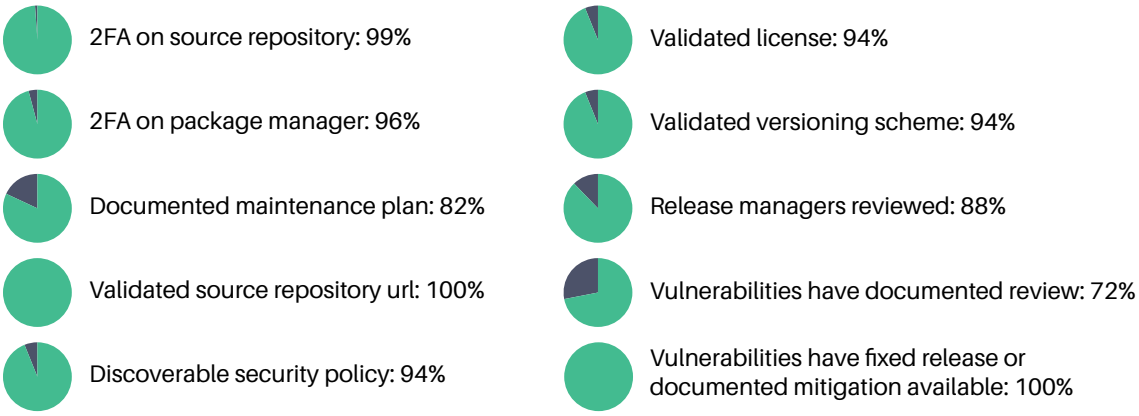
## Tidelift has clarified and incentivized standardized security work for package maintainers since 2018

Tidelift has been evolving the secure package standards seen on pages 9–10 since 2018. We measure and guide maintainer effort through a series of standards, checks, and tasks that maintainers are given to complete to retain their monthly income. This has resulted in more awareness and adoption of code vetting systems.

Some of the results of that effort are shown here:

### Tidelift consistently tracks our partnered maintainers' efforts to uphold our software development lifecycle policies

Though our task completion data is ever-evolving, this is a snapshot in time of some of the compliance with our standards for partnered maintainers. Tidelift serves as the central IT or compliance office for upstream open source packages.



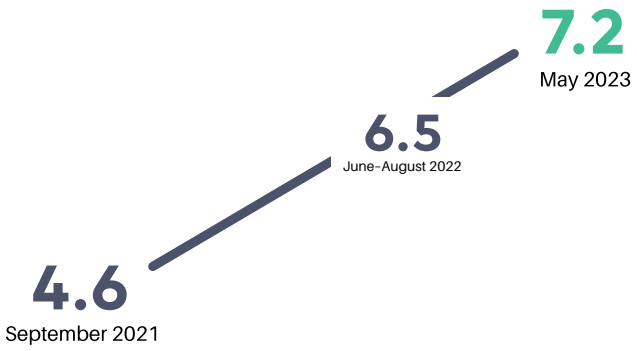

Beginning in June of 2022, Tidelift worked with the OpenSSF Scorecard project to take a deeper look at how we could improve adoption of the scorecard, and overall scores. **As expected, Tidelift's cohort for this research was already outperforming their previous scores from September 2021, as well as their peer open source packages.**

## Tidelift has driven maintainer collaboration with the scorecards project

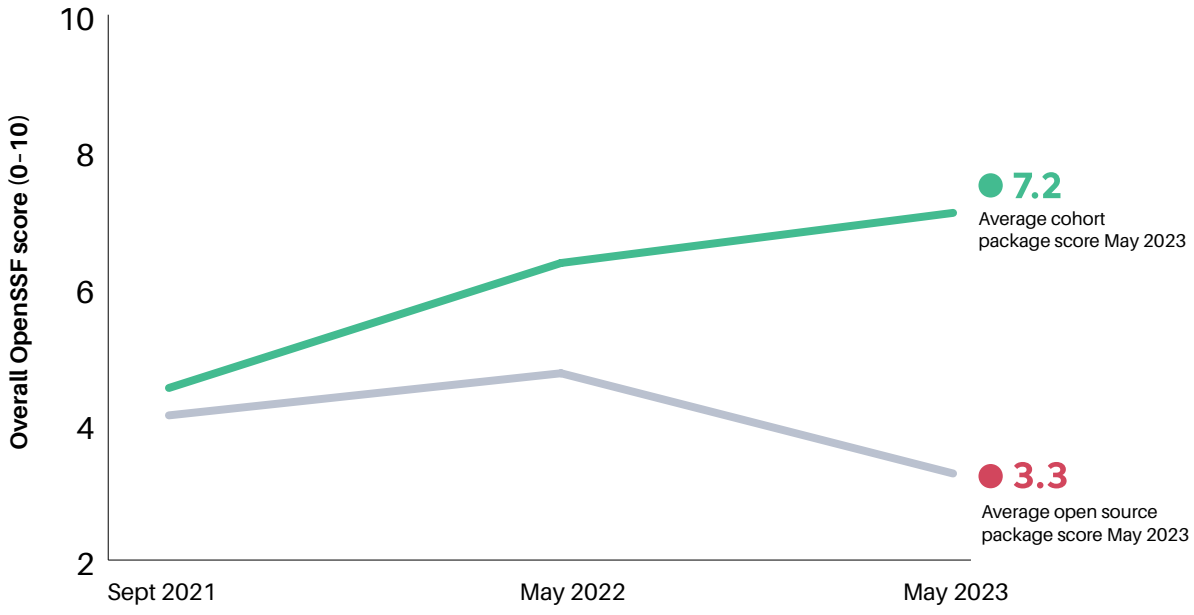
In addition to years of incentivizing adoption of a secure set of development practices, Tidelift's OpenSSF Scorecard results story shows powerful results:

### Tidelift's multi-year partnership with maintainers has increased overall scores by 57%

Through a focused effort on scorecards starting in June of 2022 to May 2023, Tidelift increased OpenSSF scorecard scores from an average of 6.5 to 7.2 (n=26), increased maintainer engagement with scorecards, and improved how the scoring is assessed.




While other packages without investment remain at lower levels of scores:




**OpenSSF scorecards over time: focused cohort vs all asessed opens source packages**

This focused cohort on the OpenSSF Scorecard gave us a window into how maintainers are thinking about adopting these kinds of industry standards, and what hurdles exist for broader adoption.

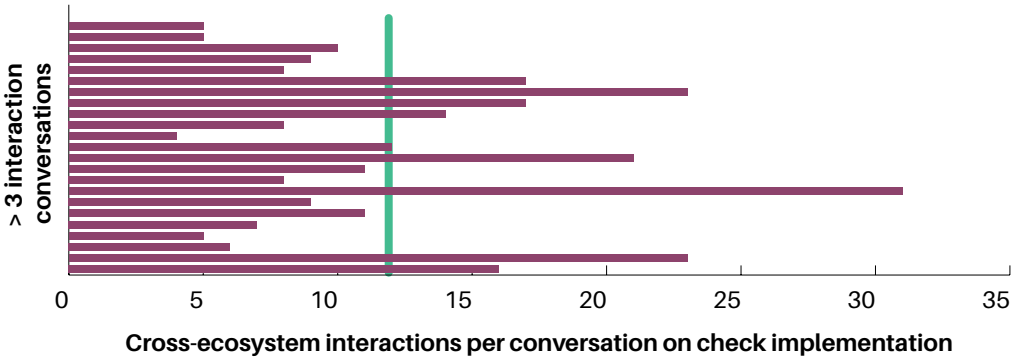
As a result of working with the scorecards team, Tidelift was able to surface data inconsistencies and improve several checks:

- Security Policy
- Vulnerabilities
- Dependency Update Tool
- Pinned Dependencies
- Branch Protection

Tidelift was also able to clearly see the challenge that checks like Code Review and Branch Protection surface when the majority of open source packages have a single maintainer working on the development of the package.

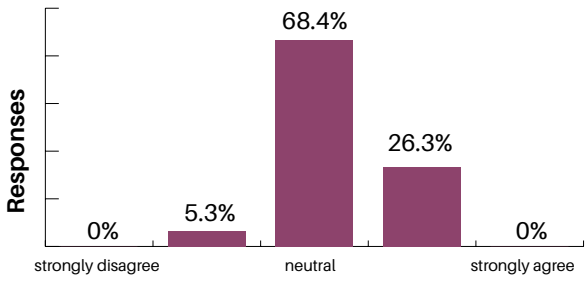
## Seeing meaningful scorecard improvements and resulting outcomes requires conversations and incentives

We saw an average of **12 cross-ecosystem interactions per discussion** on how to complete work for scorecard checks




> ***Security is something that is usually hard to prioritize. Clear suggestions of what to do, paired with funding to go and do it, is the ideal combination to help make things happen.***
>
> *— Tidelift-partnered open source maintainer*

Most maintainers in the OpenSSF scorecards cohort found the financial inventives to be neutral or better with the additional workload required



## Maintainers went further to improve specific checks when we created space for conversations and provided incentives

Though the cohort was focused on understanding and improving 5 checks, nearly 1/3 of the cohort went on to improve an additional set of checks and 72% of cohort participants reported improving other packages that they maintain as a result of this focused effort.

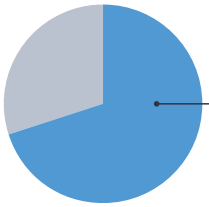

## Notable scale achievements

- Cohort maintainer released a **template for a Python package with a secure project host and package repository configuration** as well as a **recurring view of Python scores over time to increase visibility and adoption**
- Cohort maintainer requested and secured wide access for **all Apache projects to have access to a dedicated GitHub action for understanding and improving scores**
- Partnered maintainer released a Maven extension for **validating and collecting checksums of all artifacts during execution** and **pushed this value upstream to all of Maven**
- Partnered maintainer released a **Gradle plugin for addressing binary artifacts** in this key ecosystem build tool

> ***There is value here, both in making the current status visible and in teaching other maintainers. It's mostly a matter of bandwidth, since as a maintainer there are also so many other things to do.***
>
> *— Tidelift-partnered open source maintainer*

## Maintainers in the cohort applied scorecard practices to additional projects beyond the pilot

When asked 'Have you applied any of the practices that the scorecard checks measure to other packages that you maintain?' maintainers report that they are motivated to do so when incentives are in place.



**70%** Have implemented scorecard checks on other projects during the paid pilot

Out of our cohort of 26 packages, **there is a potential total of nearly 2,500 additional packages that these maintainers also work on** — increasing impact across Java, JavaScript, and Python.

## The OpenSSF scorecard needs additional refinement, and maintainers are now actively participating to improve it

The scorecard as implemented today does not allow for enough nuance to assess meaningful security practices — and outcomes — across all ecosystems and types of packages. Still, maintainers are seeing value in it and eager to participate in making it even better.

**36** **Comments**

**12** **Pull requests**

**Maintainer contributions to the public scorecards project during the pilot study**

> *The monetary incentive is important, but the metrics need refinement.*
>
> *— Tidelift-partnered open source maintainer*

> *Some checks are obvious and were already fulfilled or we complied right away. Others don't really apply or generate a lot of false positives for little benefit, so we don't bother.*
>
> *— Tidelift-partnered open source maintainer*

> *...After all, the goal is maximum security, not getting and displaying a good mark.*
>
> *— Tidelift-partnered open source maintainer*

## As a result of Tidelift's work, packages are being developed with more security practices in place, and software supply chain security is increasing

Our sustained partnership has also positioned us well to take the learnings over the last year and improve tooling and processes for yielding continuous improvement in critical open source packages:

- Developing security solutions for single maintainer packages
- Making smarter, more frictionless dependency management tooling to overcome the overwhelming noise of false positive CVE reports
- Continue to improve scoring to focus more on measuring outcomes over inputs, and build in the affordances needed for ecosystem differences and package type differences
- Measuring and strengthening more of the early build practices of package maintainers
- Working with maintainers to understand and scale the security practices they have in place today that aren't measured in the current OpenSSF Scorecard checks

FUTURE GROWTH

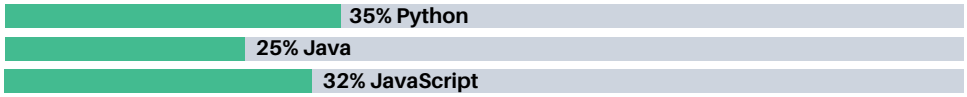
# This model is ready to scale

We can now measure investment, outcomes, and impact for open source supply chain security. With more economic investment, we can scale out further to critical packages.

Tidelift's partnered maintainer community today is covering many critical open source packages. More than 160,000 repositories depend on our lifted packages today.

## Tidelift's partnered maintainer coverage across most critical independently maintained open source packages

There are many packages that score highly on OpenSSF's criticality score that have a vendor, such as Amazon, VMWare, or Red Hat standing behind them. For the dependency graphs underpinning those packages, developed and secured by independent maintainers — Tidelift is a scalable, proven solution.




**Percentage of packages scoring 0.70 or higher according to OpenSSF Criticality Scores that have a Tidelift partnership in place, and more secure development practices assurances as a result**

Today, Tidelift's model can deliver 2FA enablement, security policy setup, vulnerability review, release manager access verification, and dependency management when a maintainer signs on to join. We have thousands of packages across hundreds of maintainers that have signed a contract and are awaiting income to deliver on secure development work beyond what they may be doing on their packages today.

As we look ahead, we want to expand how we work with maintainers, continuing how we clarify and incentivize secure build processes and moving more into token permissions, fuzzing, signing, and a range of practices oriented to reduce malicious source code injection supply chain attacks.

**We have a cohort of maintainers ready to move out into key ecosystems and provide security audit and upkeep services, including paired code reviews — beyond the packages they themselves maintain.**

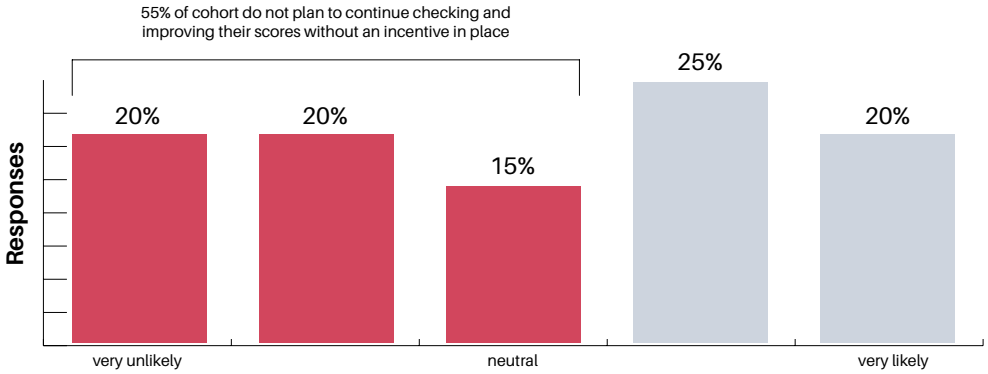
CONCLUSION

# Improving security outcomes for open source is a networked, people-centered problem

With the clarity, incentives, and relationships that Tidelift has created with independent open source maintainers, we can force multiply the practices and outcomes leading to even more secure, better open source. This will continue to drive innovation — safely — for industry and government alike. **Tidelift is a solution for proactively maximizing secure outcomes while minimizing risks.**

## Incentives and community will be key to continuing this work and securing the open source supply chain

We asked 'How likely would you be to **continue** checking and improving your OpenSSF score, were there no incentives in place?' Maintainers are telling us that this work **must be incentivized**.




**Likelihood of continuing scorecard work on package that we studied, with no incentives in place**

TIDELIFT.COM

TIDELIFT