The Tidelift guide to securing your open source dependencies

August 2019

TIDELIFT

CONTENTS

INTRODUCTION 3

WHAT YOU'LL LEARN IN THIS GUIDE 5

INTRODUCING MANAGED OPEN SOURCE 5

THE TIDELIFT SUBSCRIPTION 8

STEP 1: IDENTIFICATION 9

STEP 2: RESOLUTION 11

STEP 3: PREVENTION 14

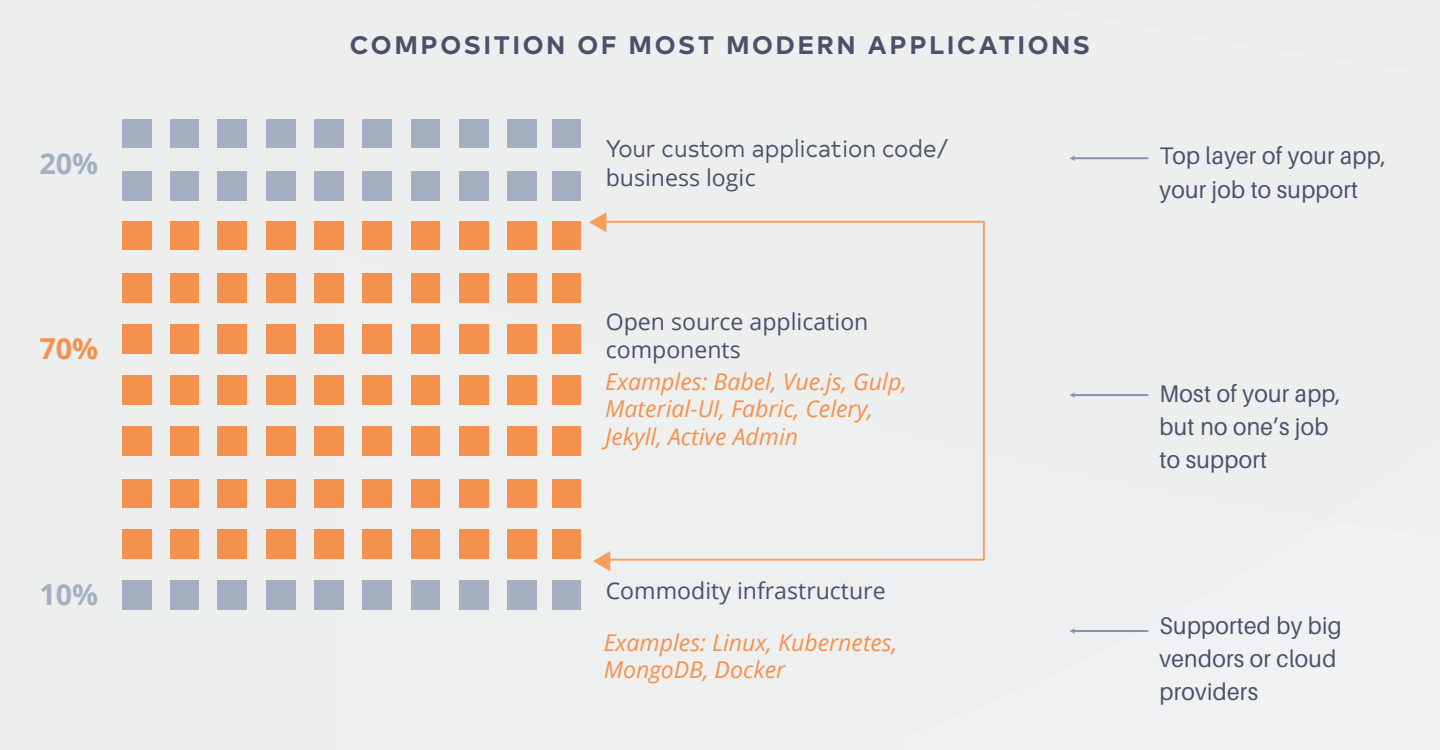
ABOUT TIDELIFT 16

INTRODUCTION

Corporate application development teams are increasing their reliance on open source as they look to bring products to market faster and focus developers' time on creating differentiated solutions, rather than recreating software components.

Ample research, including Tidelift's own, shows that most applications are built on a foundation of 70 percent or more open source code. And for new apps, the number is often higher. All told, 92 percent of professionally developed applications today depend on hundreds or even thousands of open source components.

FIGURE 1



Like all software, open source packages can contain errors, some of which can be exploited. Unpaid open source project maintainers often don't have the time to apply security fixes as quickly as corporate users would like. There have even been high-profile cases of maintainers inadvertently handing over control of their projects to malicious actors, as happened with the event-stream npm package vulnerability.

Cybercriminals recognize that unmaintained open source components represent a promising attack surface and have increased their focus on open source in recent years. IBM found that between 2006 and 2015, the average time between a vulnerability being reported to being exploited decreased from 45 to 15 days.

The industry has responded with a number of open source security scanning and alerting tools. Even governments are taking action, with the European Union's Free and Open Source Software Audit program as perhaps the most visible of these efforts.

But in a world where community-led open source components make up the foundation of nearly every application, scanning and alerts only get us part of the way there.

To see this, let's recall when development teams controlled the majority of their production code—either because they wrote it themselves or they acquired it from a known supplier with a service level agreement. Back then, security alerts by themselves added a ton of value because they enabled professional development teams to take one of two direct actions to remediate:

1. Fix it if it was in your code.
2. Contact the supplier in whose code the threat resided and demand a fix.

When you're working with open source components, however, merely knowing that a vulnerability exists in one of the thousands of direct and transitive dependencies (or dependencies of dependencies) in your application is useful. But you know what would be even more useful?

To actually have the vulnerability *fixed* in a timely, professional fashion.

Recent estimates suggest 10 percent to 20 percent of open source packages that are commonly in use by corporate teams aren't actively maintained. These aren't obscure components—roughly 20 percent of dependencies in boilerplate React, Angular, and Vue applications, for example, go unmaintained. And no less than 80 percent of maintained packages have no vulnerability disclosure policy, and therefore no mechanism to receive security reports and address them.

If a package is unmaintained, then by definition there is no one around to fix it when a vulnerability appears. And if there is no vulnerability disclosure policy in place, well, that opens the door for the dreaded zero-day exploit to rear its ugly head.

All this adds up to a state of open source security that leaves much to be desired.

WHAT YOU'LL LEARN IN THIS GUIDE

This guide details how managed open source ensures that the components you use to build applications consistently adhere to security best practices and how this reduces your risk and allows you to stay focused on your own application development.

INTRODUCING MANAGED OPEN SOURCE

Think back to before cloud computing, when to launch a SaaS app you'd need to rent space from a hosting facility near an Internet POP, buy and install servers and networking gear to ensure connectivity, backup, and failover, and then configure all of the software you needed on the equipment. When something went wrong, you'd drive or fly to the hosting facility, swap it out, install software updates, and so on.

Today, you simply tap a couple of buttons or run a script and your favorite cloud provider manages everything else for you.

Yet when it comes to the thousands of open source components that modern apps rely on, application development teams still do the heavy lifting themselves—or worse, the lifting doesn't get done at all. Let's consider just a few of the ways your team wastes time managing your open source dependencies rather than developing your product:

- Staying up to date with the latest bugfix versions.
- Moving to a new major version of a framework or library.
- Dealing with bugs or security issues related to an unmaintained dependency.
- Handling requests from your legal department to list every package you're using, along with their licenses.
- Documenting everything you use for your security team and addressing live vulnerabilities.

Our latest survey finds that these concerns are pervasive. For example, nearly 60 percent of respondents say that moving to a new major version of a framework or library presents a challenge. Over 50 percent report that adapting to bugs or breaking changes in an updated dependency presents a challenge (see Figure 2).

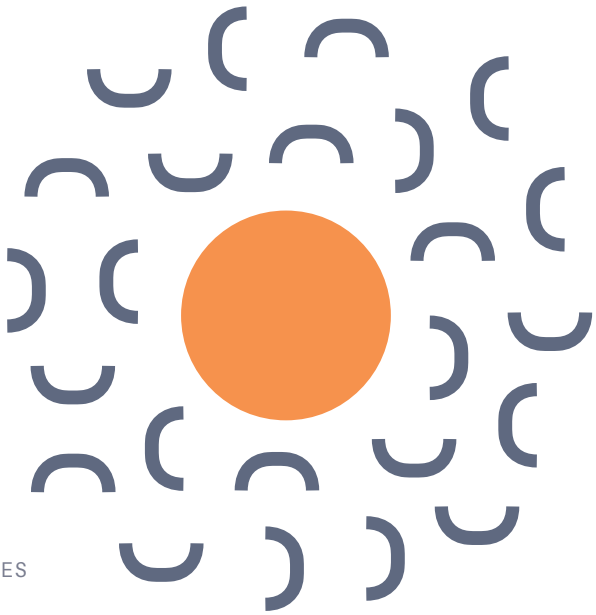
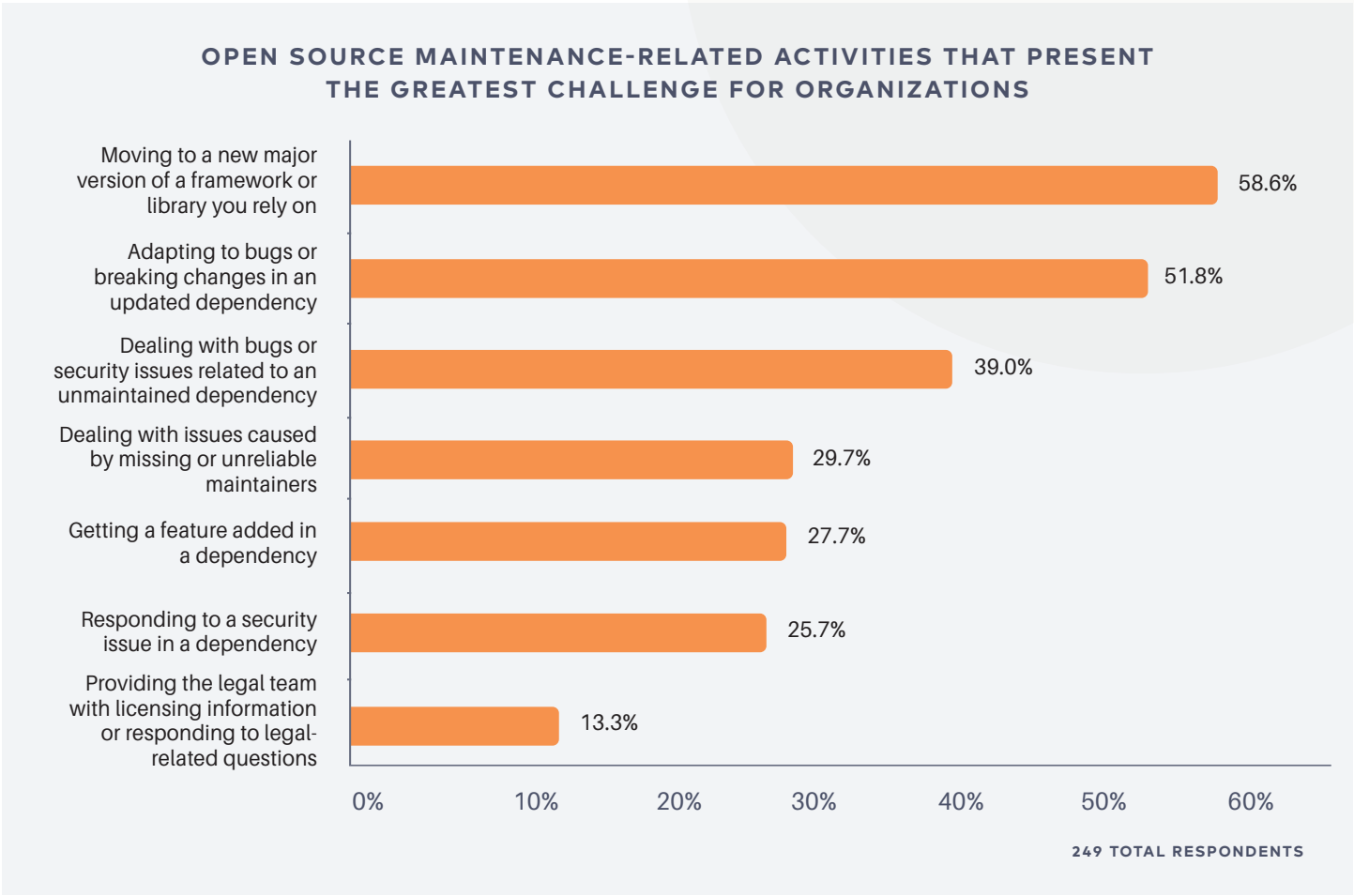


FIGURE 2



Chances are these are not the places where you'd prefer your team to be spending time. And it's certain they'd rather be doing more creative work.

Fortunately there's a revolution happening today in the way apps are built. Just as cloud computing upended the way you approach application hosting—by turning it over to your favorite cloud hosting provider—you can now outsource open source management to the experts.

WHAT IS MANAGED OPEN SOURCE?

Managed open source is a new model that allows application development teams to speed up development and reduce risk by outsourcing the management of open source components to the experts who create and maintain them.

Managed open source helps application development teams maximize their use of open source packages by ensuring they are up to date, secure, and well maintained. It also provides assurances like support under a service-level agreement and intellectual property indemnification. The end result? Better maintained software at a lower cost, because all the participating commercial users share the cost of paying the maintainers behind each open source package.

With managed open source, individual organizations no longer have to choose between bearing the costs of verifying and maintaining all of the open source they use or going without that verification. Managed open source gives organizations all the capabilities they expect from commercial-grade software for the full breadth of open source they use.

Managed open source significantly improves the security options available to enterprise development teams. Figure 3 illustrates how the Tidelift approach to dealing with vulnerable open source packages compares with those of other providers.

FIGURE 3

TIDELIFT VERSUS TRADITIONAL OPEN SOURCE SECURITY APPROACHES

	Security action	Traditional Open Source Security	Tidelift	What this means for you
1	Continuous security scanning of the OS packages in your stack	✓	✓	→ Gain baseline and ongoing view of security of the open source code you consume → Make more informed choices about the new packages to add
2	Vulnerability alerts for packages you use, including release number/branches	✓	✓	→ Enables hot patches to your app to protect users → Before Tidelift, you're still holding the bag for the long term fix
3	Pay maintainers to address security issues—proactively and reactively	✗	✓	→ Only Tidelift removes the burden of dealing with insecure open source packages from your development team by paying maintainers to address them at the root → This frees you to focus on adding features to your app

THE TIDELIFT SUBSCRIPTION

The Tidelift Subscription is a managed open source subscription backed by the creators and maintainers of community-led projects used by enterprise application development teams. With the Tidelift Subscription in place, you spend less time managing open source dependencies and more time building your own software.

Subscribers get help identifying the open source software in their apps and all associated dependencies, understanding potential issues with components, getting resolutions to those issues from Tidelift's network of open source maintainers, and choosing the best and most reliable packages to include in their apps.

Tidelift's managed open source subscription helps organizations use open source more effectively, while paying maintainers for the immense value they create.

The Tidelift Subscription covers all of the open source components your organization uses, helping you monitor 3.3 million packages across 37 different ecosystems. In addition, more than 1,000 projects that are pivotal to commercial application development, including Apache Struts, Joda-Time, Vue, Babel, Material-UI, Gulp, Mongoose, and Nokogiri—with more added every day—are directly backed by the maintainers themselves. With Tidelift's open source coverage, and backing of open source maintainers, subscribers get access to the most complete knowledge about open source dependencies in the industry.

With Tidelift's open source coverage, and backing of open source maintainers, subscribers get access to the most complete knowledge about open source dependencies in the industry.

The Tidelift Subscription provides a variety of actionable views into all this data to keep your team moving fast. These include an overview of security vulnerabilities, licensing issues, and technical concerns across dependencies; at-a-glance metrics that help developers gauge how package updates impact their applications; and recommendations on when to upgrade key frameworks and libraries.

This functionality is backed up by Tidelift's network of open source maintainers, who work to resolve security, maintenance, and licensing problems on your behalf, freeing up your developers' time.

STEP 1: IDENTIFICATION

Bottom line is that, when it comes to security, scanning and alerts are table stakes. Because your production apps rely on open source, you need to know immediately when there's a vulnerability. The first step to securing your open source is to generate a central inventory of the libraries and other components your applications use.

The Tidelift Subscription provides the tools development teams need to continuously catalog and understand the open source software their app depends on.

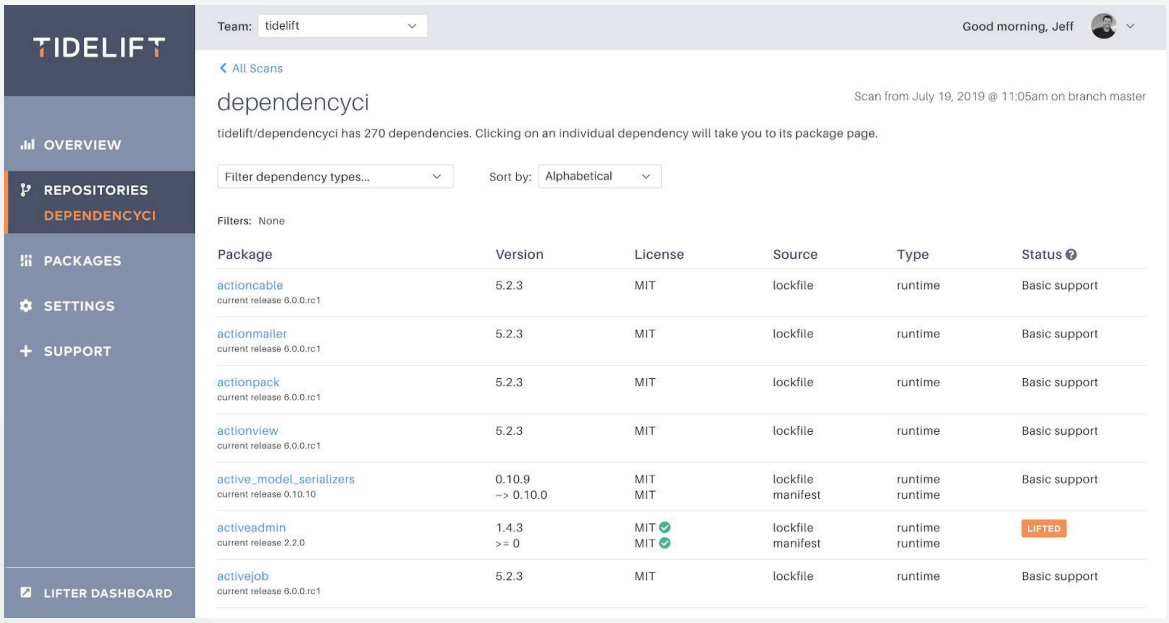
We simply scan your package manager files and extract the dependency list from them.

1. If you use GitHub.com, we call GitHub.com APIs to get files in your repository, extracting only package.json and similar package manager files. Alternatively, we offer a simple API to upload your package manager files from any CI system.
2. We parse the package manager files to extract your list of dependencies and store them on our platform.
3. We regularly scan for changes on your master branch as well as on pull request branches.

This allows us to produce an always up-to-date dependency catalog that your entire team can use to know where you stand.

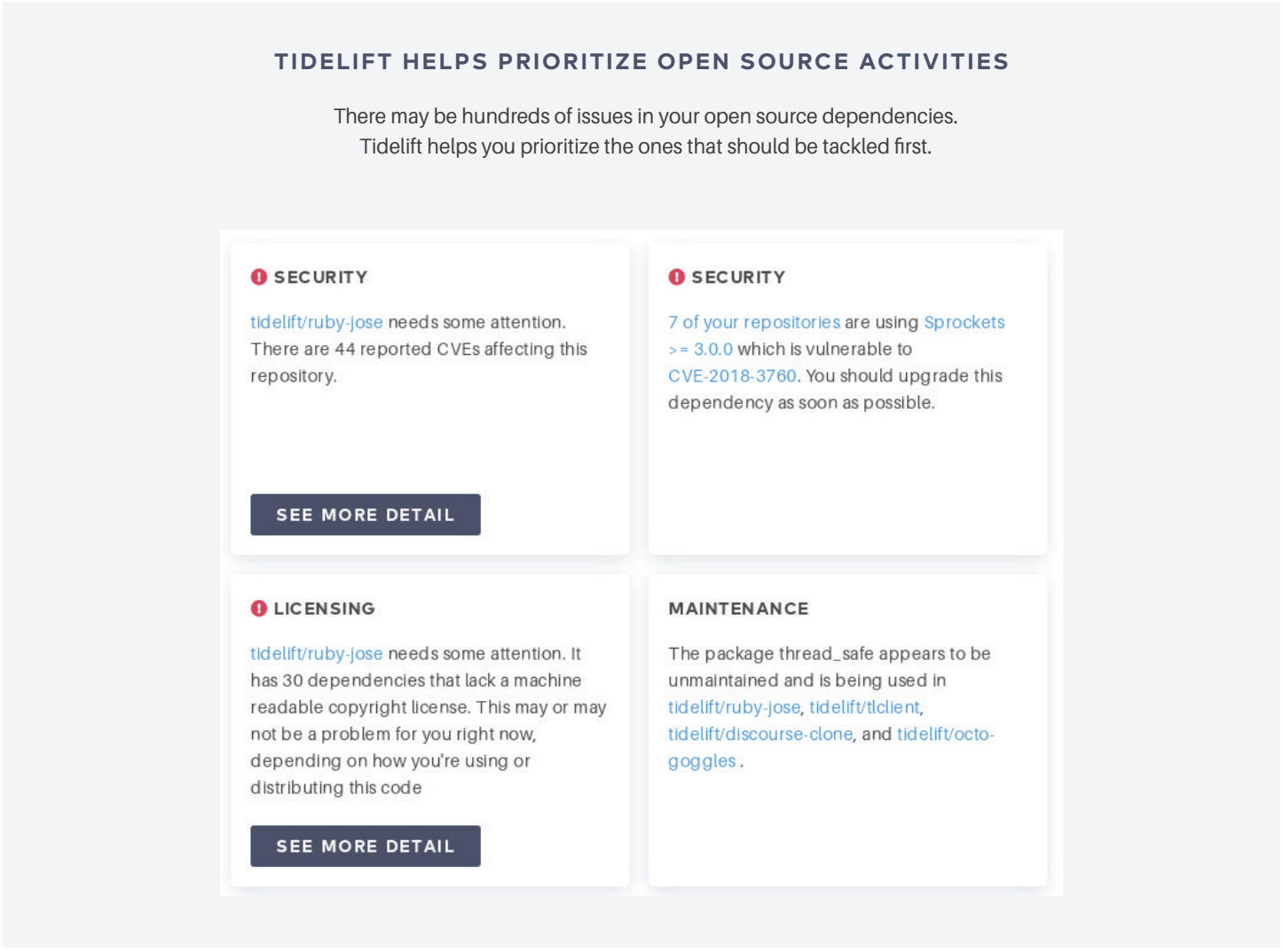
FIGURE 4

SAMPLE DEPENDENCY CATALOG



We also analyze this data to make it even more actionable. The Tidelift overview summarizes key metrics to track over time across all your projects: security vulnerabilities, licensing concerns (including a license your policy may prohibit), unmaintained packages, and outdated packages.

FIGURE 5



STEP 2: RESOLUTION

Almost every development effort today starts by tapping into the library of existing open source components. Without a managed open source platform like the Tidelift Subscription, relying on a distributed army of maintainers when you need a fix yesterday can place a significant drag on development speed. That's because, until Tidelift, maintainers of popular projects (who are usually volunteers!) were often not adequately incented to keep up with all of the requests coming their way.

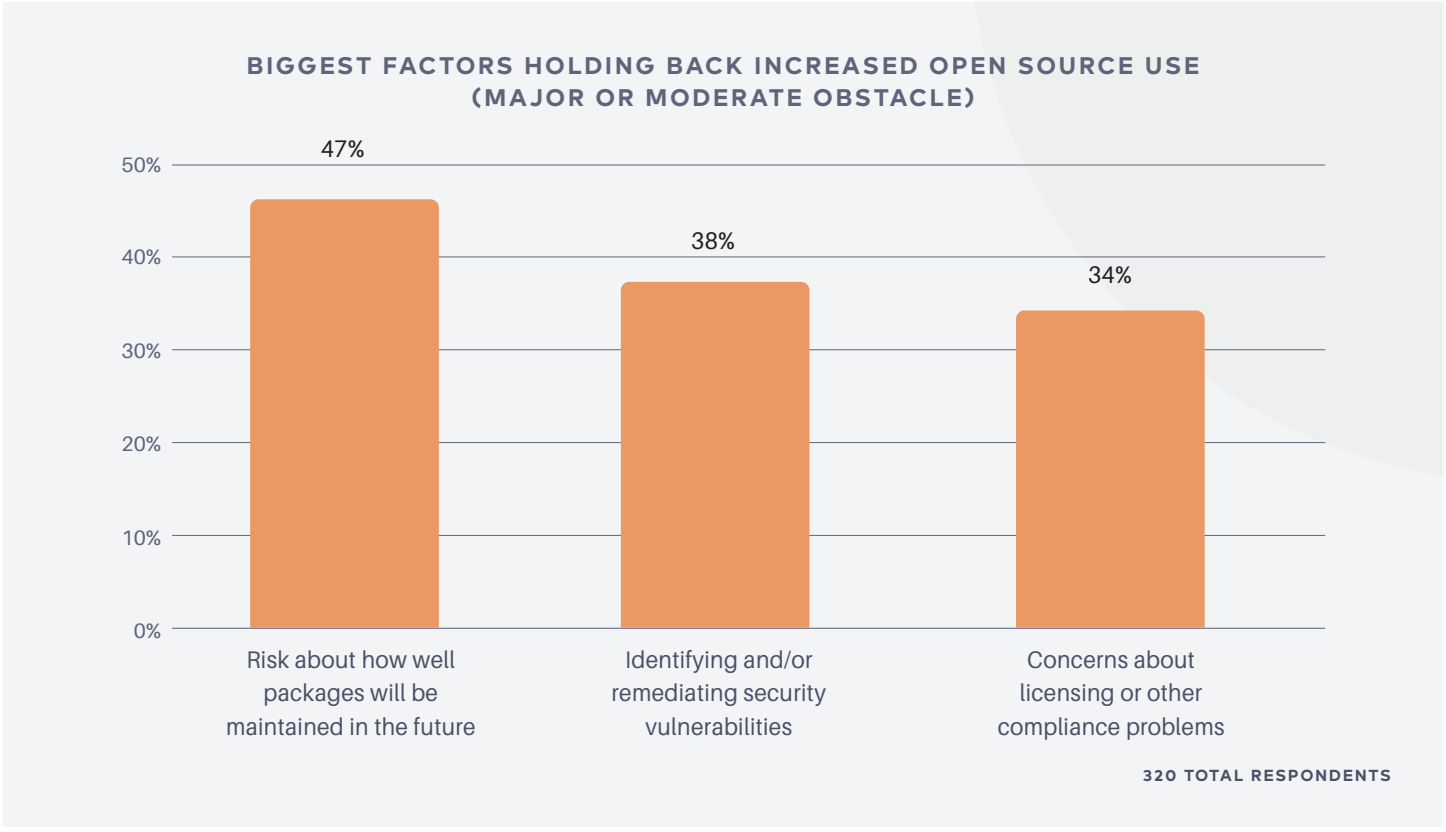
When an open source vulnerability emerges today, you probably go through the following steps:

1. Look upstream to see if the maintainer is working on a fix. If they are, in many cases it will be months before a fix gets merged.

2. Start considering your long-term solution. Normally the options are replace the package or fork, fix, and maintain it yourself.

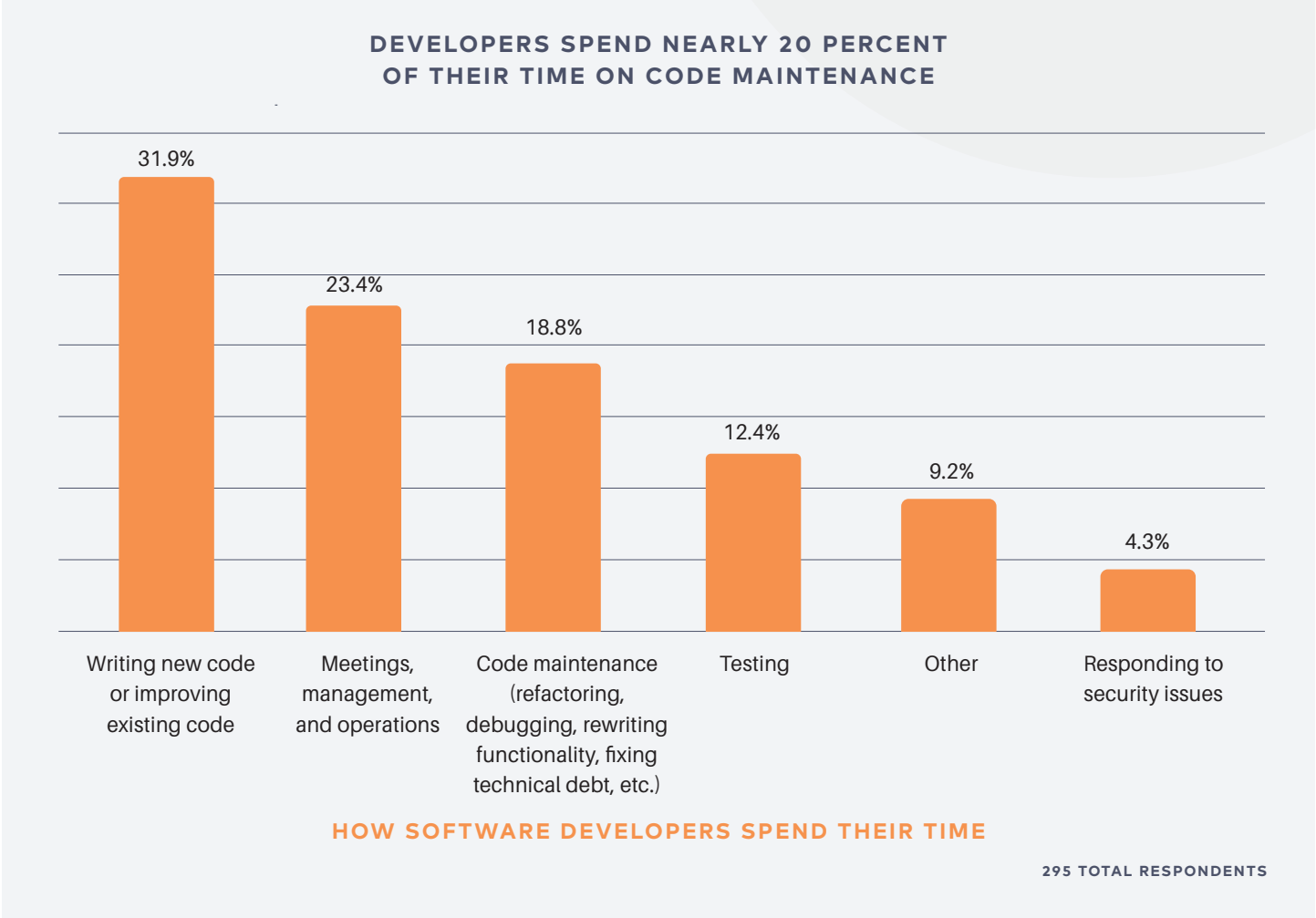
Based on our most recent survey, these concerns are holding developers back from expanding their use of open source, as shown in Figure 6.

FIGURE 6



Our survey also found that technologists who use open source packages spend an average of nearly 20 percent of their time on code maintenance, with more than 25 percent of that time devoted to maintaining their applications' open source dependencies.

FIGURE 7



The biggest organizations are particularly weighed down by this task. Teams with more than 500 developers spend fully one-third of their code maintenance time wrangling open source dependencies.

With a managed open source subscription, enterprise developers can outsource package security resolution to Tidelift. Unlike other solutions that leave you on the hook to contact maintainers and often to work on a long term fix, Tidelift works with maintainers on your behalf to quickly resolve vulnerabilities.

Once a vulnerability emerges, we work directly with the maintainer to prioritize a fix. Depending on the nature of the work, the maintainer may introduce an intermediate branch prior to a long-term solution. In either case, the vulnerability is addressed at its root. This prevents you from having to make the hard choice of re-engineering to another package or "in-sourcing" a fork of the original package.

Tidelift also provides a mechanism to avoid hair-on-fire zero-days. Typical small open source projects have nowhere to report security vulnerabilities confidentially. This results in zero-days when vulnerabilities are reported via tweet or public GitHub issue, resulting in a frantic global scramble by every organization that uses that software.

Tidelift helps our maintainers post secure reporting instructions on their project pages, and then we walk them through a coordinated disclosure process to ensure there's a fix available at the same time the vulnerability is announced.

STEP 3: PREVENTION

Tidelift equips maintainers of the packages you use with the tools to implement robust security policies, and pays them to stay on top of issues using your subscription money, which goes directly to the maintainers of the exact packages you use.

As a first step, maintainers of supported packages implement a vulnerability-reporting link and policy to their README, SECURITY.md file if using GitHub, or on the project website. Maintainers can create their own process or they can use the Tidelift reporting page and process.

Noted analyst Scott Crawford from 451 Research believes a vulnerability reporting policy should be considered table stakes for any company with a public footprint. We agree, and think this advice applies equally to open source packages.

Tidelift maintainers agree to follow a number of other important security best practices, as detailed in Table 1.

TABLE 1

SECURITY PRACTICES FOLLOWED BY TIDELIFT'S NETWORK OF MAINTAINERS

Security practice	Description	How it helps you
Responsible disclosure	Process wherein security issues are disclosed only after a period of time that allows for the vulnerability or issue to be patched or mended. We ask maintainers to add a vulnerability-reporting link and policy to their README or project website. They can optionally use the Tidelift reporting page and process.	Your customers and users entrust their sensitive data to you, so you need time to address vulnerabilities in the open source tech you use. This ensures you have that time.
Be responsive to security issues	Maintainers have multiple competing priorities. We ask that they put security issues at the front of their priority queue.	New features are always great, but even a minor security hole can scuttle an entire project. By working with maintainers to put security at the top of the list, we're looking out for you.
Tell us about new vulnerabilities	If they are not using the Tidelift reporting page, we ask maintainers to enter any new vulnerability into the maintainer dashboard when the vulnerability becomes public. We'll then notify any affected subscribers.	You get notified of security problems faster.
Use two-factor authentication and strong password policies	2FA provides an extra layer of security to make sure that people trying to gain access to an online account are who they say they are. First, a user will enter their username and a password. Then they will be required to provide another piece of information: something you know, something you have, or something you are.	In the past, packages have been replaced with malicious code after their maintainers' accounts were hacked.

ABOUT TIDELIFT

Tidelift makes open source work better—for everyone

Through the Tidelift Subscription and in direct partnership with maintainers, Tidelift is a single source for proactively managed open source components and professional assurances around those components. Tidelift makes it possible for open source projects to thrive, so we can all create even more incredible software, even faster.

ABOUT THE TIDELIFT SUBSCRIPTION

A managed open source subscription backed by creators and maintainers

The Tidelift Subscription manages your dependencies for you, delivering all of the capabilities you expect from commercial-grade software, for the full breadth of open source you use.

- **We provide the tools** you need to continuously catalog and understand the open source software that your application depends on.
- **We partner with and pay the open source community maintainers** of the exact packages you use, to ensure they meet the standards you require.
- **We address issues proactively,** not only scanning for new security, licensing, and maintenance issues, but also working with our participating open source maintainers to resolve them on your behalf.
- **We help you measure and improve your open source dependencies' health**—which improves your app's health—and give you a short list of high-impact steps your team can take to improve them even more.
- **We add commercial assurances that don't come for free** with open source packages, like intellectual property indemnification and support under a service level agreement. You expect these guarantees from proprietary software, and you should get them when using open source as well.

Request a demo and learn more: tidelift.com/subscription