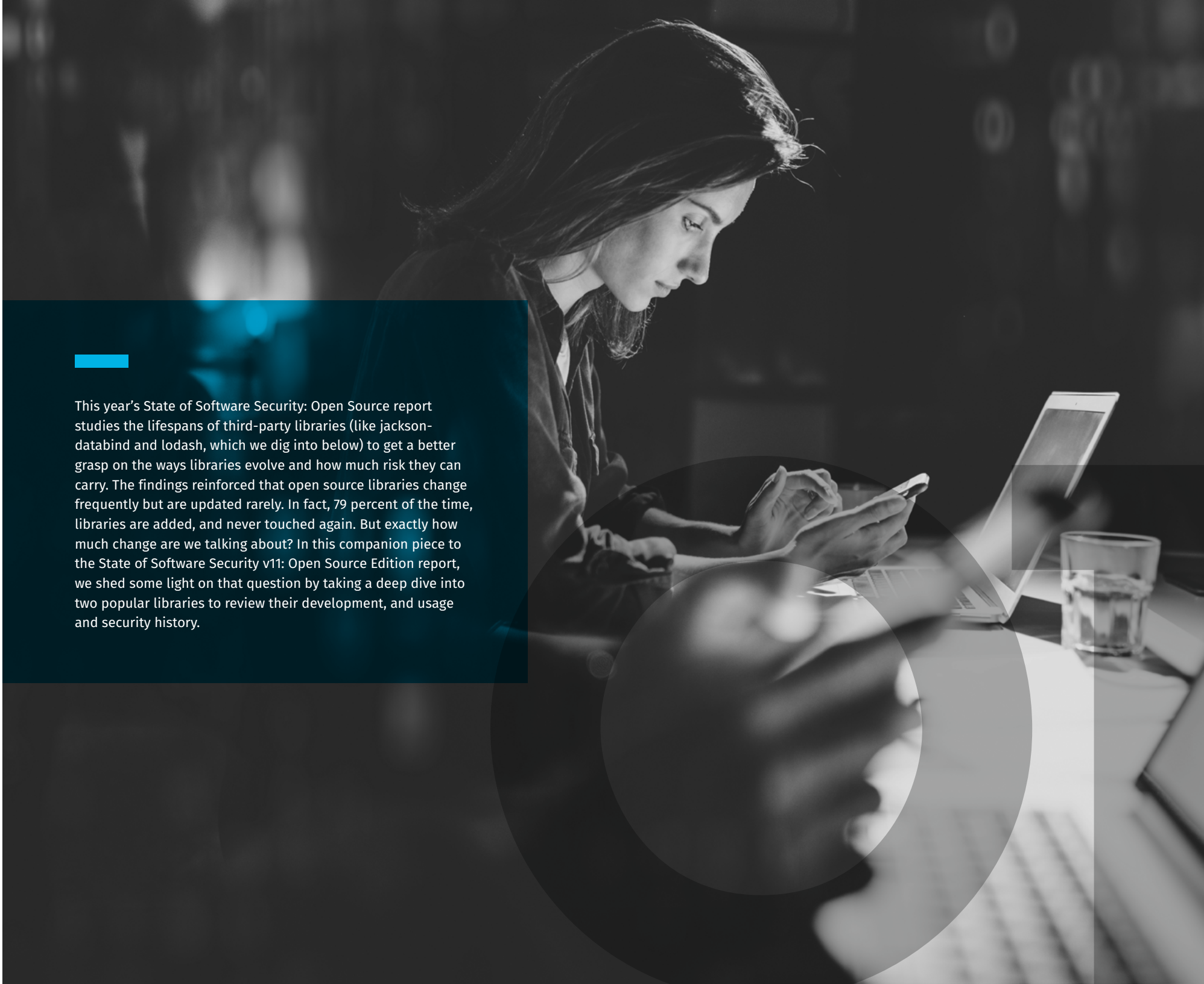
VERACODE

STATE OF SOFTWARE SECURITY:
VOLUME 11

Open Source Edition

THE LIFE AND TIMES OF THIRD PARTY SOFTWARE: LIBRARY CASE STUDIES



This year's State of Software Security: Open Source report studies the lifespans of third-party libraries (like jackson-databind and lodash, which we dig into below) to get a better grasp on the ways libraries evolve and how much risk they can carry. The findings reinforced that open source libraries change frequently but are updated rarely. In fact, 79 percent of the time, libraries are added, and never touched again. But exactly how much change are we talking about? In this companion piece to the State of Software Security v11: Open Source Edition report, we shed some light on that question by taking a deep dive into two popular libraries to review their development, and usage and security history.

jackson-databind

The Jackson Data Processor is a suite of Java libraries focused on the processing of JSON data. The jackson-databind library in particular converts JSON structured data into Java objects, a highly useful function that finds it used in nearly seven out of 10 applications that include any Java libraries. But because it handles the transmuting of data – including potentially user data – into code objects, it is ripe for vulnerabilities. In this library's lifetime, there are nearly 6,000 commits, dozens of releases, and nearly 70 separate vulnerabilities. While its commit history goes back to nearly 2012, we focus on the time period in which we see its use by Veracode customers – starting in late 2016[1].

Another more sinister drumbeat is that of vulnerabilities. We see a number released starting in early 2018. A heartening reaction to this is seeing new, non-vulnerable releases committed and Veracode customers adopting those versions at a steady rate. One other hero worth mentioning in this story is a single developer, GitHub ID cowtowncoder aka Tatu Saloranta. Tatu has done the vast majority of work on this critical project, including addressing many security issues.

While we might often think that releases are the effort of hundreds or sometimes thousands of developers, that isn't always the case. Tatu Saloranta's lone effort is proof positive that we can't simply assume libraries are being thoroughly maintained by a group, and thus security is carefully considered. And in cases like the jackson-databind library, the data shows us that it's simply taking too long to update these libraries once a patch is released for a vulnerability, adding to the risk already lingering from libraries with minimal (or nonexistent) upkeep.

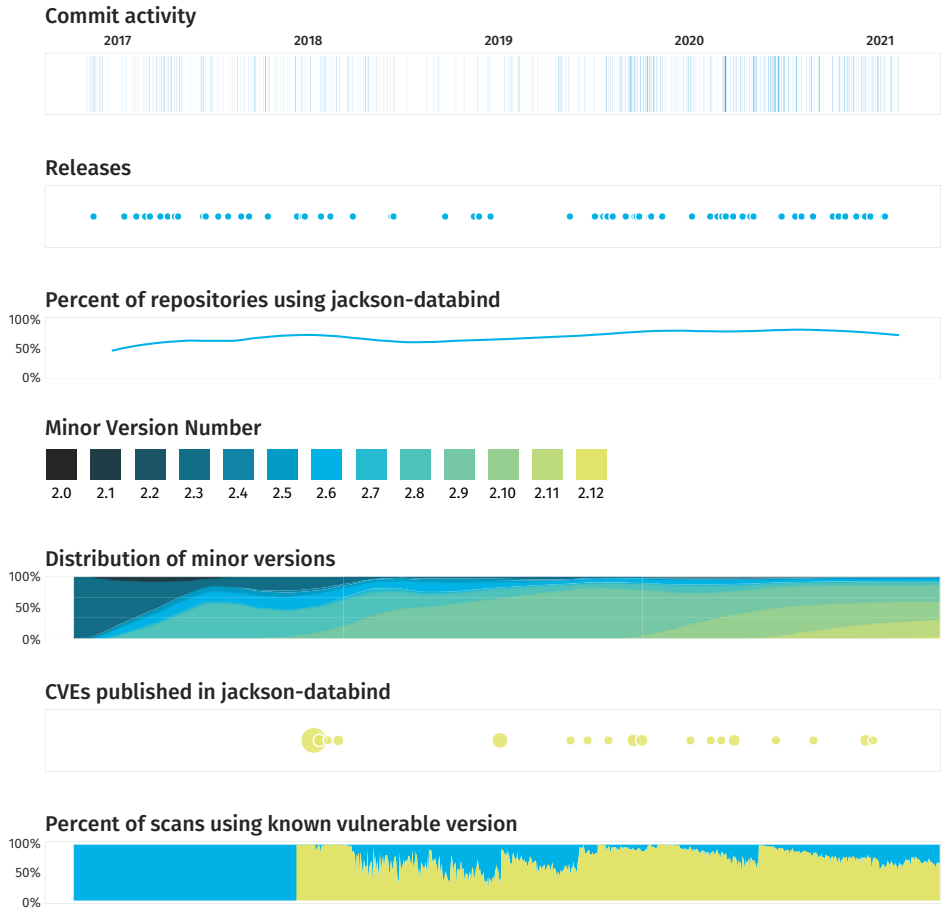


Figure 1: *A library case study: jackson-databind*

[1] *That's not to say Veracode customers weren't using it before 2016, but rather that's when Veracode started collecting the usage data.*

lodash

We'll move from an extremely popular Java library to an extremely popular JavaScript library, one that appears in more than 80 percent of repositories that use any JavaScript libraries. lodash aids in the handling of base JavaScript types and provides functionality that didn't make an appearance in the main language until ES5 and ES6, released in 2009 and 2016 respectively[2].

lodash is unusual for such a popular library. Most popular JavaScript libraries are extremely small, sometimes only a few lines long, and usually implement a single or handful of functions. lodash's codebase is much larger, implementing more than 325 functions and containing 35,000+ lines[3] of code.

Because lodash handles data and provides extensions to core language types, it is also ripe for vulnerabilities. While it has fewer total vulnerabilities than jackson-databind, these vulnerabilities affect a larger number of versions, with only the latest version (4.17.21) free from known vulnerabilities (for now). We visualize the development of this library in the same way as we did jackson-databind below and see some of the same patterns, with a few key differences.

When the first set of vulnerabilities was released in June of 2018, lodash already had functional versions (specifically 4.17.10) that were not vulnerable, though the majority of applications were using vulnerable versions (4.17.5 and prior). Like jackson-databind, lodash is largely the work of one developer – however, by mid-2018, the library had entered what could be called "maintenance mode" having not seen more than a patch-level update since that time.

This was especially problematic when a prototype pollution vulnerability (CVE-2018-16487) was published in February of 2019, affecting all versions before and including 4.17.11. Version 4.17.12 wasn't released until July of that year fixing two other similar vulnerabilities, dooming developers to rely on vulnerable versions.

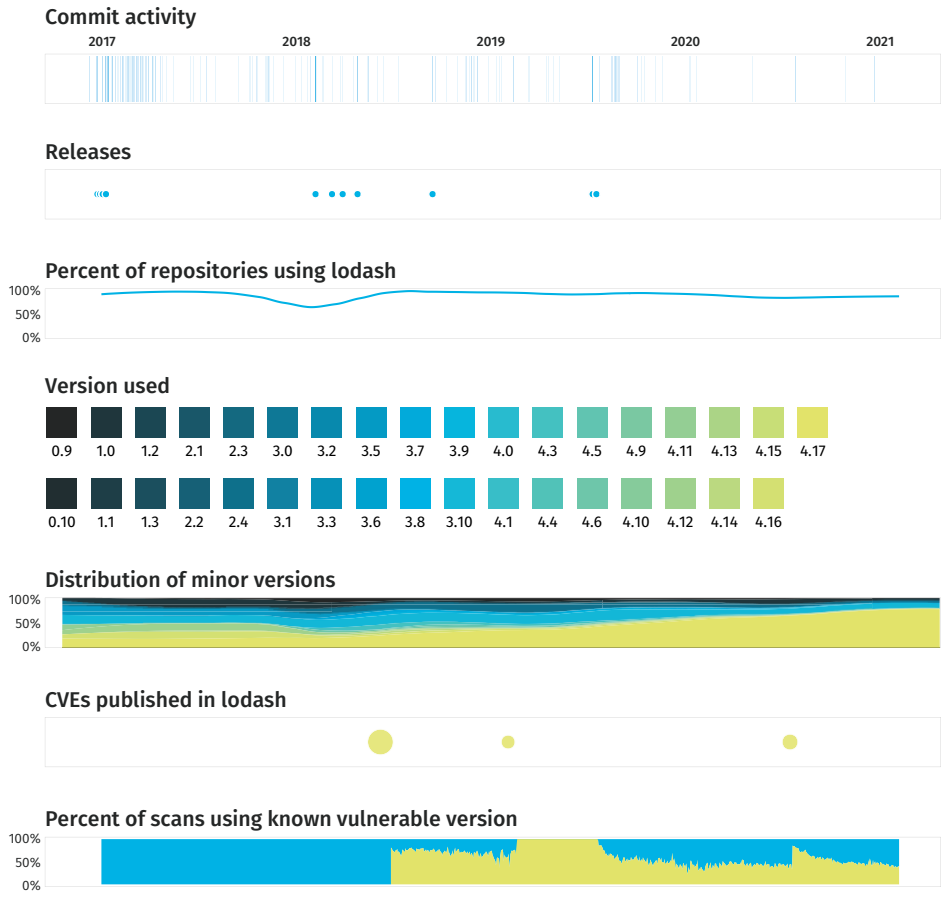
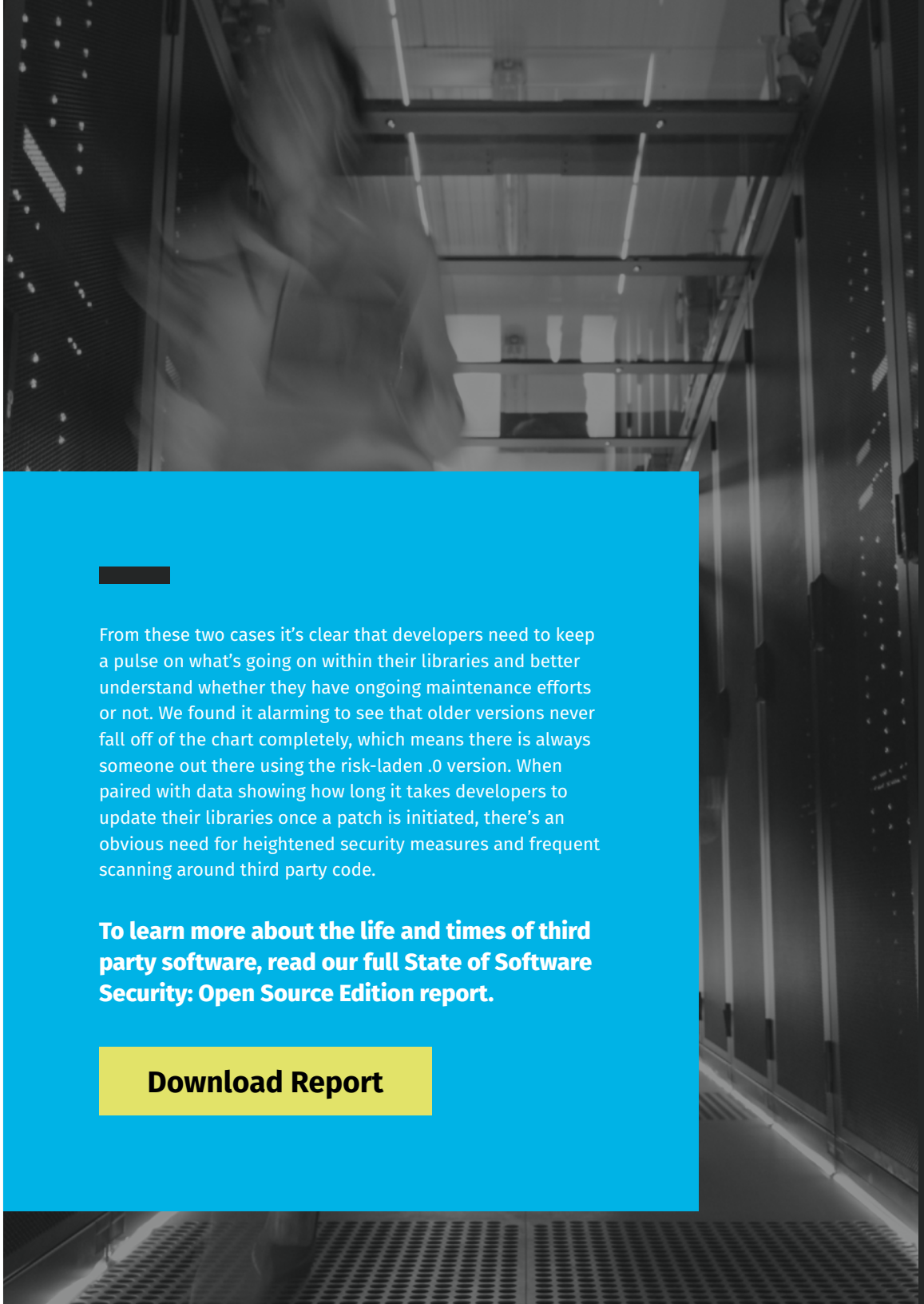


Figure 2: *A library case study: lodash*

[2] *Primarily functional style programming primitives like map, reduce, and filter. lodash still provides functions which operate on JavaScript objects, whereas in the base language this is only available for Arrays.*

[3] *Though mini-fying down to just a few kilobytes.*



From these two cases it's clear that developers need to keep a pulse on what's going on within their libraries and better understand whether they have ongoing maintenance efforts or not. We found it alarming to see that older versions never fall off of the chart completely, which means there is always someone out there using the risk-laden .0 version. When paired with data showing how long it takes developers to update their libraries once a patch is initiated, there's an obvious need for heightened security measures and frequent scanning around third party code.

To learn more about the life and times of third party software, read our full State of Software Security: Open Source Edition report.

Download Report

VERACODE

Veracode is the leading AppSec partner for creating secure software, reducing the risk of security breach and increasing security and development teams' productivity. As a result, companies using Veracode can move their business, and the world, forward. With its combination of automation, integrations, process, and speed, Veracode helps companies get accurate and reliable results to focus their efforts on fixing, not just finding, potential vulnerabilities. Veracode serves more than 2,500 customers worldwide across a wide range of industries. The Veracode cloud platform has assessed more than 14 trillion lines of code and helped companies fix more than 46 million security flaws.

www.veracode.com

Veracode Blog

Twitter