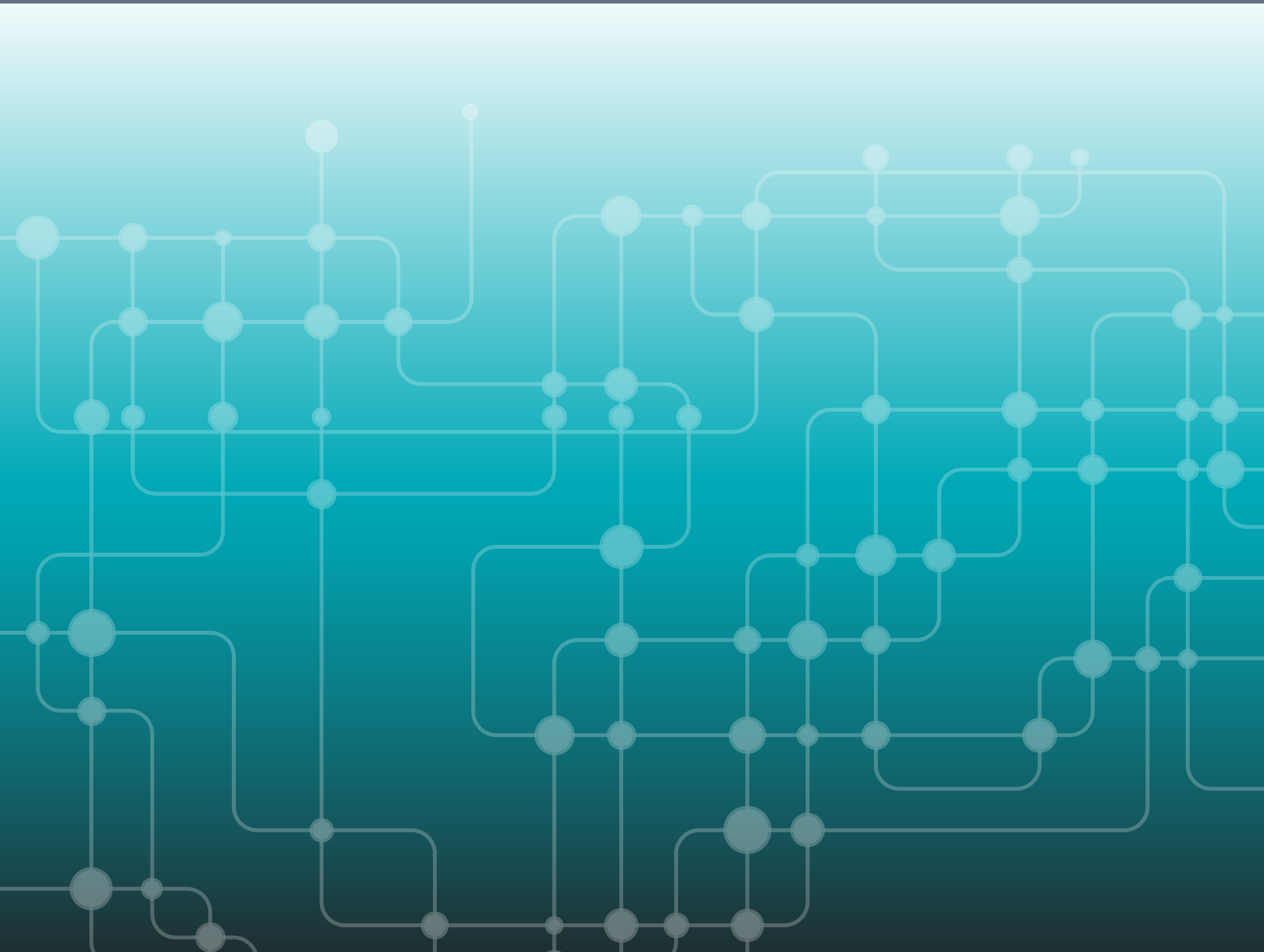
WHITE PAPER

2025 Open Source Risk in M&A by the Numbers



The numbers at a glance

Best practices for firms involved in merger and acquisition (M&A) transactions include audits of the target's code whenever software is a significant part of the value of a deal. Expert third parties that perform multiple types of analyses on the code can help purchasers better understand the technology and capabilities they're buying and identify potential legal, security, and quality issues.

*In 2024, our audits found **open source in 99% of transactions** we audited, with a mean number of **2,778 components** discovered per engagement.*

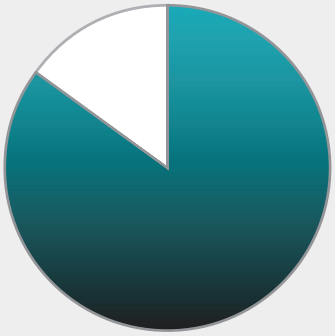
Today, open source components and libraries form the backbone of nearly every application in every industry. The reasons are straightforward: Using open source speeds development, drives innovation, and lowers costs, all critical in today's agile software world. The Black Duck Audit Services team conducts open source audits—as well as analysis of software security and quality—on thousands of codebases (the code and associated libraries that make up an application or service) for its customers each year. Our audits are primarily done in support of M&A transactions to provide customers with comprehensive, up-to-date Software Bills of Materials of the open source, third-party code, web services, and APIs used in their applications, and to enable a view into risks associated with the components.

The 2025 "Open Source Security and Risk Analysis" (OSSRA) report presents analysis from an examination of the anonymized data from about 1,000 commercial codebases audited as part of hundreds of M&A transactions in 2024. Industries represented in the report include automotive, big data, cybersecurity, enterprise software, financial services, healthcare, the Internet of Things, manufacturing, and mobile apps.

The OSSRA report presents data on codebases—the code and associated libraries that make up an application. The typical transaction that involved a Black Duck audit in 2024 included four codebases. As this paper focuses on M&A transactions, data is presented by transaction, meaning the software being acquired via the transaction. For example, a mean number of 2,778 open source components were found per transaction.

The Black Duck Audit team audits thousands of codebases for our customers each year, with the primary aim of identifying software risks during M&A transactions.

Acquirers in M&A deals want to understand what risks may be associated with the software they're acquiring. For the open source elements of a target's code, the concerns are with the licensing, security, and quality of the specific components used in that software. The audits found open source in 99% of transactions, and a mean number of 2,778 open source components were discovered per transaction. On average, 70% of this "proprietary software" was in fact open source and third-party code. Given the time-to-market, cost savings, and development advantages of leveraging open source components, it's no surprise that companies continue to rely heavily on open source as part of their software development process. But the large number of discrete components speaks to the challenge of tracking it all.



85%

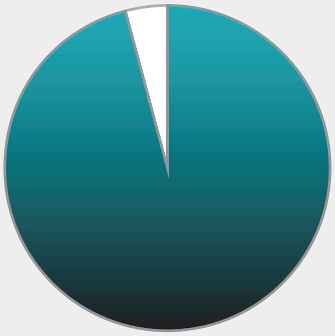
of transactions included components with license conflicts

The vulnerabilities and license compliance issues discovered in the codebases were almost as pervasive as open source itself. In 2024, 85% of transactions included components with license conflicts. The most common conflict was related to the Creative Commons Attribution ShareAlike 3.0 license, closely followed by ShareAlike 4.0.

"Snippets"—lines of code that have been copied and pasted into source code—are quite frequently found by the Black Duck Audit team. These snippets are often taken from the popular blog site Stack Overflow, which automatically licenses all publicly accessible user contributions under Creative Commons Attribution ShareAlike. Unfortunately, the blanket license also covers code snippets posted on the site. We say "unfortunately" because Creative Common licenses are not intended for software, with Creative Commons explicit about this in its FAQ: "We recommend against using Creative Commons licenses for software." The CC-SA license can be read in some situations as having a similar "viral" effect (that is, any work derived from a copyleft-licensed work must also be licensed under the same copyleft terms) as the GNU Public License (GPL), a license that also turns up in audits, and can become a concern from a legal standpoint.

Unpatched software vulnerabilities are one of the biggest cyberthreats organizations face, and unpatched open source components in software add to security risk. Ninety-six percent of transactions included unpatched, vulnerable open source, with a mean of 519 vulnerabilities per transaction. Ninety-six percent of the transactions also contained open source with at least one high-risk vulnerability. High-risk vulnerabilities are those that have been actively exploited and already have either a documented proof-of-concept exploit or classification as a remote code execution vulnerability.

Our audits showed that jQuery was the #1 component containing vulnerabilities. Sixty-nine percent of the audited codebases contained the jQuery component. It should be noted that jQuery is not inherently insecure. In fact, it is a well-maintained open source library with a large community of users, developers, and maintainers. But according to the audits, jQuery was the component most likely to have vulnerabilities, even though all the jQuery vulnerabilities listed in the 2025 OSSRA report have available patches. To reiterate, this is not meant as an indictment of the quality of jQuery, but rather of the processes that companies use to monitor and patch vulnerabilities. It is important for users of jQuery—and indeed users of all open source—to be aware of the potential security risks associated with the use of older versions of software, and to take steps to mitigate those risks.



96%

of transactions included vulnerable open source

*In **98% of transactions**, the software included components that had **no new development activity** in the last two years.*

Of the customer transactions conducted by the Black Duck Audit team in 2024, 98% of the software audited included components that had no new development activity in the last two years. Ninety-seven percent had open source more than four years out-of-date, that is, using open source with newer versions available—often with many newer versions available.

Besides contributing to security risk, getting too far behind in versioning means that software is falling behind functionally and is becoming more difficult to maintain. That is because the more versions that need to be leapfrogged when upgrading a component, the harder it is to integrate the latest version. Thus, falling behind can be costly and risky when an upgrade becomes necessary to eliminate a vulnerability or for any other reason.

Best practices in the use of open source software require development teams to understand which components and associated licenses are in their code and what obligations result from their use of that open source. It also requires monitoring for newly discovered vulnerabilities and keeping up-to-date with the latest versions. All this requires knowing what open source components are in a codebase. However, tracking open source manually can be an impossible task for any organization.

Customer audit engagements in 2024 conducted by the Black Duck team found a huge number of open source components (a mean number of 2,778 components per transaction), each with versions, vulnerabilities, and licenses that need to be tracked.

This volume of components suggests the importance—if not absolute necessity—of automated open source management. At that scale, a company cannot rely on manual processes, and an acquirer cannot assume that a target is comprehensively tracking the open source it uses.

The reality is that many companies, particularly smaller ones, don’t have the necessary processes and tools to manage their developers’ use of open source; it’s simply not a high priority to the typical startup. Although we’ve seen improvement since we started reporting 10 years ago, it is still the case that most targets are unable to produce an open source Software Bill of Materials easily and routinely. When they do, it is rarely accurate and never complete. And even for open source such companies are aware of, they don’t tend to track security vulnerabilities, as evidenced in the statistics in this paper.

Software composition analysis for M&A due diligence

Knowing what open source code is in a company’s codebase is crucial for properly managing its use and reuse, ensuring compliance with software licenses, and staying on top of patching vulnerabilities—all essential to containing business risk. From an M&A perspective, a code audit enables a buyer to understand risks in the software that could affect the value of the intellectual property and the remediation required to address those risks, as well as to plan out a roadmap going forward. Savvy sellers may employ an audit proactively to avoid surprises in due diligence, particularly given the amount of unknown open source in a typical company’s code.

This is why an open source audit can be invaluable for companies wanting a better understanding of the code’s composition. Using techniques known as software composition analysis (SCA) and a range of sophisticated tools, expert auditors comprehensively identify the open source components in a codebase and flag legal compliance issues related to those components, prioritizing issues based on their severity. The audit identifies known security vulnerabilities that affect the open source components, as well as information such as versions, available patches, and the state of a component’s development activity. It also provides indications as to the sophistication of a target’s software development processes. Open source is so ubiquitous today that if a company isn’t managing that part of software development well, it raises questions as to how well it is managing other aspects.

If you’re on the buy side of a tech M&A transaction, an open source audit should absolutely be part of the software due diligence process. Acquirers need to identify problematic open source in the target’s code before the transaction terms and integration plans are set, and a trusted third-party audit is the best way to get a deep, comprehensive view. Prospective sellers should prepare for questions about the composition of their code and how well they have managed open source security and license risk. Absent good open source management processes, proactive sellers can prepare for an acquisition by having their software audited in advance.

Open source license compliance risk

Like all software, open source components are governed by licenses that vary in terms of rights, obligations, and restrictions. Failure to comply with open source licenses can put businesses at risk of litigation and could compromise their intellectual property. Generally, as part of a definitive agreement, sellers need to represent that they have the rights to any software they are using (and there may be even more explicit representations and warranties regarding open source). Yet few sellers are totally "clean" when it comes to open source license compliance.

Organizations can manage and comply with license requirements only if they can identify the open source components and confirm that the use of those components is consistent with the terms of the applicable license. Just as with security vulnerabilities, it's impossible to manage license compliance risk without identifying all the components in the software.

Open source license compliance remains critical

Based on Black Duck audit data, investors in all software verticals should be concerned about open source licensing and the potential risk of litigation or threat to their intellectual property rights due to failure to comply with an open source license. Black Duck analyses indicate that the 20 most popular licenses cover approximately 98% of the open source in use. But whether the software in question uses one of those popular licenses or some variant, the license matters.

*In **76% of transactions**, auditors found **open source with customized licenses**, or components freely available on the internet but with no discernible license at all.*

License risk arises when software includes components with licenses that conflict with the with the way the overall codebase is licensed. For example, the GPL is an extremely common license that often governs components in commercial software and requires distributors to make source code available. But commercial software vendors typically do not offer to provide source code or complete source code, which creates a conflict with that license.

Sometimes an open source component has what Black Duck refers to as a "custom license" in which the developer created their own licensing language or added language to create a variant of a standard license. Customized open source licenses might place undesirable requirements or limitations on the licensee and will often require legal evaluation for possible IP issues or other implications. For example, the JSON license is based on the permissive MIT license, but the JSON license adds the distinction that "the software shall be used for good, not evil." The ambiguity of this statement leaves its meaning up to interpretation, posing a particular concern in M&A scenarios where acquirers and their counsel need to weigh any legal risk. And there are many examples of developers being similarly "creative" with licenses.

Black Duck audits conducted in 2024 found that 76% of transactions included open source with customized licenses or no license at all. If third-party code is used without a license, this raises legal concerns. In the U.S. and many other jurisdictions, creative work—including software—is placed under exclusive copyright by default. Unless there's a license from the copyright holder that grants permission, no other party can use, copy, distribute, or modify the software without the risk of litigation.

Broken down by industry, 100% of codebases in six sectors had open source license conflicts.

- **Computer Hardware and Semiconductors**
- **Cybersecurity**
- **Internet and Mobile Apps**
- **Manufacturing, Industrials, Robotics**
- **Retail and eCommerce**
- **Virtual Reality, Gaming, Entertainment, Media**

The Energy and Clean Tech sector had the lowest percentage of codebases with open source license conflicts, at 60%.

Permissive and reciprocal licenses

Open source licenses fall on a spectrum from permissive to reciprocal. Permissive licenses place minimal obligations on companies that redistribute the associated software. By contrast, reciprocal (also known as "copyleft") licenses require the licensee to make any improvements or enhancements available to the public under similar terms. In some cases, the entirety of the work that incorporates the licensed software, even a small portion of it, may fall under the reciprocal obligation.

Permissive licenses

Permissive open source licenses generally require only that the licensee attribute the original portions of the licensed code to the original developers both in the code and in documentation. Of course, to provide such an attribution, the licensee must be aware that they're using the licensed code, so it is no surprise that most targets are not fully compliant with permissive licenses. While this is generally considered a lesser risk than, say, distributing GPL-licensed code, it is still an issue that most acquirers will want to address in their plans to meet their own corporate standards, typically stricter than those of most targets.

Reciprocal licenses

On the other end of the spectrum, codebases containing reciprocal licenses are quite problematic for an acquirer. Many of these licenses require associated code to be made available to the public under the same license. A licensee that violates a reciprocal license could be at risk of litigation and may be required to disclose all the source code of the application. Such issues often come to light when a licensee is acquired in an M&A transaction, and acquiring companies will want to remediate either before or after close.

IP compliance risk introduced by AI coding tools

Arising with the use of Generative AI-powered coding tools are questions around ownership, copyright, and licensing of the generated code. For example, a class-action lawsuit filed against GitHub, Microsoft, and OpenAI claims that GitHub Copilot—a cloud-based AI tool that offers developers autocomplete-style suggestions as they code—violates both copyright law and software licensing requirements. The lawsuit further claims that the code suggested by Copilot seems to, at times, copy/paste licensed code without attribution, copyright notice, or adherence to the original licensing terms. Microsoft offers some level of indemnification when the tool is used under certain guidelines, but user companies still need to be wary as many other such coding tools do not provide any protection.

The Copilot case highlights the legal complexities surrounding the use of AI-generated code. For software developers, refraining from using AI-assisted coding tools until the issue is resolved by legal or government decision is obviously the safest way to avoid an action for license or copyright violations, but the reality is that many developers continue to use such tools. An open source audit as part of software due diligence will highlight instances where an AI has "copied" code from open source projects and associated risks, just as with human-written code.

Open source security risk

Large organizations may manage hundreds to thousands of software assets, ranging from mobile apps to cloud-based systems to legacy systems running on premises. That software is typically a mix of commercial off-the-shelf packages and custom-built codebases, both of which contain volumes of open source components.

As noted earlier, the Black Duck Audit team found open source in 99% of transactions in 2024. Here's the reality: If an organization builds, sells, or simply uses software, it's safe to assume that the software contains open source.

In 96% of transactions in which we were involved, we found unpatched open source vulnerabilities, with a mean of 519 vulnerabilities uncovered per transaction. Ninety-six percent of the transactions had at least one high-risk vulnerability. "High-risk" indicates that a vulnerability has been actively exploited, has documented proof-of-concept exploits, or has been classified as a remote code execution vulnerability.

Open source projects usually issue small updates at a much higher frequency than the average commercial software vendor. When these updates contain security updates, companies need to have a strategy to adopt them rapidly. But because open source updates need to be "pulled" by users, many companies consuming open source components don't apply the patches they need, exposing their business to the risk of attack and their applications to potential exploits. This is understandable—these companies typically aren't even aware of the outdated components in their code, let alone looking for patches.

The data indicates that development teams may be struggling with the dynamic nature of open source security risk, especially with the increase in open source use. An open source component with no known vulnerabilities doesn't necessarily stay that way a year, month—sometimes not even a week—later. So unlike with licenses that don't change with great frequency, up front vetting is insufficient to ensure code security.

The struggle is illustrated by organizations being startlingly behind in using the latest version of any given open source component. Ninety-seven percent of our 2024 M&A transactions included open source components more than four years out-of-date. Ninety-eight percent of the transactions included components that were not the current version—exposing the codebases to security risks and other issues. Firms acquiring companies, undergoing mergers, entering joint ventures, or even managing underlying technology supply chains must also consider the risks they are potentially inheriting.

Summary

The thousands of audits conducted by the Black Duck Audit team have consistently revealed that almost every codebase contains open source code. As documented in this paper, 99% of transactions audited in 2024 included open source components.

The Black Duck Audit team generally audits codebases from software-heavy companies, as opposed to enterprises that use software to support their business. The primary value of software companies is their proprietary code. The ratio of open source to total code in ***their*** codebases, while still quite high at 70%, is eclipsed by how much open source is used in large enterprises. The figures cited by analysts such as Forrester, which generally look at enterprise IT groups for their reports, consistently find that open source often comprises up to 90% of new codebases, and that over 90% of IT organizations use open source software in mission-critical workloads.

> *With the magnitude of open source usage comes risk, due primarily to organizations lacking the needed tools and processes to recognize what—and how much—open source is in their internal and public-facing applications.*

Failure to comply with open source licenses can put businesses at significant risk of litigation and jeopardize their ownership rights to their software. Perhaps more importantly, especially in the context of M&A transactions, failure to comply with open source licenses more likely to result in demands for additional acquirer protections, pre-close remediation, escrows, purchase price reductions, or deal delay. Eighty-five percent of transactions in 2024 involved components with license conflicts. Seventy-six percent had open source with customized licenses or no discernible license at all. In other words, license issues are pervasive in the M&A deals we see, and they are potentially costly. At minimum, acquirers need to be mindful of these risks and reflect remediation in their post-close plans.

By identifying open source code and third-party components and licenses, an open source audit can alert your firm to potential legal and security issues in an M&A transaction. With an open source audit, you can also

- Avoid surprises
- Mitigate legal exposure
- Understand risks that may affect software asset values
- Resolve potential issues before they affect the transaction
- Build appropriate protections into the deal terms
- Plan integration and remediation

For more information, download the 2025 "Open Source Security and Risk Analysis" report to learn more about Black Duck audits.

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.