SYNOPSYS®

WHITE PAPER

When It Makes Sense to Perform an Open Source Audit

Phil Odence, General Manager, Black Duck Audits, Synopsys

We in the Black Duck® Audit Services team generally advise performing an open source audit for any transaction in which the value of the software assets is a significant part of the deal value. We refer to those as "tech transactions." It's quite reasonable to take that perspective with a grain of salt as it comes from the vendor of audit services. But while self-serving, it comes from an intimate understanding of the risks involved. It's an admittedly conservative position. The purpose of this paper is to outline the thinking behind our view that even organizations with more risk tolerance should adapt it into their own policy.

The fundamental issue

Software creation today is much more about assembling pieces and parts than working from scratch. Hardware went through the same transition in the days when engineers went from working with transistors and gates to employing complex functional chips like microprocessors. The software parts come from a variety of sources, but mostly (over 95% according to the Synopsys "Open Source Security and Risk Analysis" report) from the millions of open source components freely available on the internet. And this is a great thing for time to market, cost, and innovation.

The problem is that because open source is so freely available to developers, they can incorporate it themselves without any inherent need to let anyone know. And that can result in many companies literally not knowing what's in their code. There can be issues with open source code (just as with any other code) that need attention, but by and large developers are much more focused on functionality, performance, and meeting the build deadline than licensing, security, and future support.

Thus, absent adequate management and visibility, there are likely problems in the code developers use in their projects. Nearly 100% of transactions we audited contained code with at least one vulnerability, and over 75% had license conflicts.

The risks in open source

The most acute risks involve licensing and security. In the extreme, improperly licensed code can lead to lawsuits and potentially loss of proprietary intellectual property (IP). And components with known security vulnerabilities may leave software open to hackers. Such lawsuits and serious hacks are not daily occurrences, but they do happen. That's the nature of risk. On this topic, I'm reminded of my (then) four-year-old daughter who had a habit of carelessly sprinting from sidewalks through mall parking lots without looking. When confronted, she assured us, "But I didn't get hit by a car." She's still alive in part because we were able help her understand the concept of risk. Humans typically don't have an intuitive sense of risk.

More tangibly, if you are an acquirer and your company has a policy of using only code for which you're properly licensed, there's a 100% chance that you'll have to fix issues that show up after you close an M&A transaction. Wouldn't you want to know about them advance, so you can incorporate that development time into the integration plan? In our observation, most strategic acquirers are pretty serious about open source in their due diligence. Private equity investors are certainly aware of the high likelihood that any open source issues they acquire will ultimately be unearthed when they divest.

Less acute but equally beneficial to know about in advance are the operational issues one takes on with software that contains out-of-date or inactive components. A huge benefit of good open source code is that it's supported by a community that finds and fixes issues before you know you have them. On the other hand, an unsupported component can represent a maintenance burden. Developers are on their own to fix code with which they have no familiarity. They likely would have been better off writing it themselves.

Open source audits will shine a light on these issues and risks, so they are ideally performed before closing.

Evaluating the risk of not auditing a codebase

Our research clearly demonstrates a strong chance that any software being acquired will contain open source components with licensing, security, and operational issues. The question, then, is how important is it to understand them in advance? There's a cost to performing an audit, so the benefits need be weighed against the potential consequences and costs of not performing it.

The nature of the business

This paper is based on the premise that if software assets are a significant part of the valuation of the target company, an audit is called for. But what about when that's not the case? Almost any business today uses software, but a target in an acquisition may not be developing their own and only using off-the-shelf solutions. Although that software almost certainly contains open source, managing the issues is more the concern of the vendor.

There are many companies for which software is not the business, but it does enable the business. A primarily brick-and-mortar retailer or an insurance company with a significant online presence might be less likely to be the focus of a lawsuit and have less-valuable IP at risk. Still, security is likely a concern, so it's prudent to look into open source as part of the security evaluation. And any company developing software should be sensitive to quality and maintainability.

Tech transactions

If the target is a software company, perhaps selling software as a service (SaaS) or shipping a system that's software-based, it's risky to ignore open source risks. We'll explore levels and approaches to evaluating software for open source, but first it's worth considering: Are all tech transactions created equal?

SaaS/cloud solutions are very popular today, and there's an argument that SaaS solutions are less risky from a license perspective. An IP lawyer friend and I wrote a whole paper on just that subject (for the serious student). Although there's more license risk with distributed software, a key point in that paper is that most SaaS companies in fact distribute software as well—a mobile app front end is a common example. And because SaaS is often publicly exposed, security tends to be a risk area for those types of targets.

There's also an argument to made for looking at how well a company has managed open source on its own. If a company has always abided by a solid open source policy, had processes in place, and used open source management technology appropriate to the scale of its operation, there will almost certainly be fewer latent issues than in a company that has had its open source head in the sand. But this kind of discipline is rare in companies being acquired. Anecdotally, we've found few targets that are even 50% accurate in identifying the open source it is using.

Approaches

Process due diligence

Because companies have some amount of proprietary software, and virtually all proprietary software contains a significant amount of open source, the contents of the software should at minimum be a focus of due diligence. We've developed a checklist of questions about a company's practices and guidance on how to interpret the answers that can help you gauge how sophisticated a target is with respect to its open source management. For tech transactions, we recommend an audit regardless—and absolutely if the answers to these inquiries are shaky. However, it would not be unreasonable to have this conversation early in the due diligence process, before making the decision about auditing.

It's starting to be the case that even some smaller companies are using automated tools to scan their software for open source content. That they are aware enough of the issues to have taken a stab at getting visibility is a positive. On the other hand, many acquirers understand that automated tools have real limitations when it comes to auditing code, particularly compared to the expertise that comes with a professional third-party audit. Bear in mind too that tools require proper use and configuration and must be pointed at the right code. For all these reasons, savvy acquirers typically don't rely on the output of automated tools operated by the target and will want an independent audit.

In any case, part of any due diligence process should be getting the target company to disclose the open source components in its software, as well as the licenses. It's not unusual today to have these lists included in definitive agreements and represented to by the seller. But most attorneys I know would tell you that representations and warranties are a safety net and not a substitute for due diligence. And that can only be effectively done through audits.

Prioritization

Acquisition targets often have a variety of applications or products and even in-house tools that could be candidates for audit. It can add up to a mass of work. But auditing doesn't have to be all or nothing. Our team has the scale and flexibility to get it all done, but customers may make tradeoffs for budget reasons, or they may decide to leave some investigation until after closing.

It's not unreasonable to prioritize. Criteria we've seen include:

- Most popular products. It's not unusual for a target to be valued for a particular product A, but also have developed apps B, C, and D that will come along or the ride. The acquirer may not have any interest in some products, so focusing on the more important or popular products is rational.

- Current versions. While there may be a number of versions supported and in use, some customers choose to only audit the latest, as older ones are being phased out over time. It's worth scoping multiple versions though, because if the product or application has incrementally evolved, the additional work to audit the differences is often minimal.
- Focus on distributed. License risk is most acute with distributed software, so there's rationale to prioritize on-premises solutions over SaaS. Still, the security of open source is also an important risk to consider, and it must be taken into account for SaaS applications.
- Size considerations. For a smaller acquisition, if the target doesn't have a lot of customers yet, there's less risk of any sort. On the other hand, company size to some extent correlates with the size and complexity of the codebase. A full audit of a small codebase can be quick and inexpensive.
- No focus on internal tools. For the most part, internally used software is less of a risk from a licensing and security perspective. We have many clients that don't audit internal code for open source. For applications that an organization depends on though, understanding the architecture, quality, and maintainability can still be very important.
- No need to be comprehensive. Ideally, you'd like to identify all the risks in all the software, but in the end (with respect to open source or anything else) analyzing a representative set of codebases should give an indication of overall risk. For a large target with scores of apps, you may want to intelligently sample.

Hybrid approach

Acquisitions are so situation-dependent that a blend of the above can be the best answer. Maybe you find out through interviews that the target has multiple products teams with different approaches to open source; you might prioritize products from the least sophisticated. Or maybe it has a relatively new open source program that has only been implemented for the newest product, so it could make sense to focus audits on the legacy. For a SaaS company with distributed mobile front ends, maybe you audit the distributed code and only the most-popular SaaS offerings.

Our sales and technical advisors can help with the tradeoffs and work with you to meet timelines and budgets. We recommend scoping in any case, and doing so early in the process, so everyone knows what they are dealing with. Our default is to lay out a comprehensive solution, but if you provide our team with guidance in advance on timing and cost sensitivities, that can help steer toward an optimum solution more quickly. In any case, we will work with you to shape the best tradeoffs.

Summary

Virtually all software today is open-source-heavy. But most sellers in M&A transactions aren't equipped to manage the risks inherent in the open source their developers bring into their code. So open source audits should be part of any acquirer's software due diligence playbook. The perfect solution from a risk-mitigation perspective may not always be possible or practical, so as is often the case in life, tradeoffs are necessary. It's important to remember that the choice isn't binary—to audit or not to audit. Thoughtful, situation-dependent tradeoffs will yield the right solution for your circumstances.

SYNOPSYS®

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

For more information, go to www.synopsys.com/software.