LAW360®

Portfolio Media. Inc. | 111 West 19th Street, 5th Floor | New York, NY 10011 | www.law360.com
Phone: +1 646 783 7100 | Fax: +1 646 783 7161 | customerservice@law360.com

In-House Counsel Tips To Avert Open Source Software Risks

By **Marguerite McConihe and Greg Penoyer** (July 19, 2022, 6:00 PM EDT)

Used properly, open source software is an excellent tool. It saves your business time and money, enables interoperability of product platforms, and developers love it.

Marguerite McConihe

But used improperly, it can be financially and operationally devastating. For example, the statutory damages for failure to properly adhere to the open source software copyright notice can be up to $150,000 per act of infringement.

Those damages can quickly add up to serious consequences, whether preventing a sale or merger of your company or the destruction of the value of the affected products.

Another serious risk is that once open source software is used in your code base and deployed in distributed products, if your tech teams are not monitoring and applying bug fixes, known vulnerabilities become Trojan horses of opportunity for bad actors.

Greg Penoyer

The good news is that protecting your company from these types of risks is rather simple. We have outlined below key steps and processes in-house counsel should take to work with your business stakeholders to mitigate these risks.

1. Have a policy.

Establish a documented policy that is vetted and agreed upon by your legal, technical and compliance teams. Such a policy creates ground rules for the company and its developers regarding usage of open source software across its products and services. More importantly, creating and deploying a policy forces productive discussions and deliberations between different functional groups to align on concerns, goals and best practices for the company.

2. Be proactive.

You should always know where and how the company is using open source software. This can be done by providing proactive guidance prior to open source software use and periodically conducting audits as part of proper IT hygiene in advance of a need for the results.

Shortly before an impending transaction — M&A or commercial with a customer or partner — is not the time when you want to learn of noncompliance issues.

Performing such reviews strictly reactively can present a couple of challenges.

First, it typically leaves little or no time to remediate any identified issues.

Second, when dealing with third parties, reactive behavior and problematic results can suggest a sense of lack of sophistication or readiness. This may undermine confidence in the company or product and lead to a reduction in transaction revenue.

3. Build an iterative process into the product development cycle.

In many cases, it will be easier to stay ahead of open source software issues by treating open source software compliance like other legal or compliance approvals and incorporate it in whatever product development gate reviews or checks are used by your company for other discipline approvals, such as intellectual property product clearance, safety and quality.

This is helpful to force discussions early in the product development process and keeps your product management and engineering teams accountable for proper open source software usage and compliance.

4. Incorporate diligence into vendor and component selection process.

In-house counsel should also look upstream in the product development cycle to identify open source software used in hardware or software components being considered for incorporation into your company's products. Particularly in the case of procuring software or components that will be material to a product, it is critical to understand open source software exposure well in advance of integration and commercialization of that product.

And when vetting competing technologies or providers, consider the extent to which they rely on open source software and how that may affect your use of that open source software, such as imposing copyleft requirements on your own proprietary code, which can require you to publish and similarly grant free licenses to your own code.

These licensing consequences should factor into your assessment of the cost and value of such technologies and providers. You should also consider formalizing these commercial understandings and expectations in commercial agreements with third parties, such as contractor, consulting, joint development and software license agreements.

5. Make adoption as easy as possible.

Finally, a primary challenge in-house counsel may face in deploying an effective open source software program is the sheer administrative effort that the program may impose on the company's engineering teams to keep up with tracking use of open source software and compliance with the relevant licenses, including avoiding problematic licenses, and creating and publishing license acknowledgement reports.

There are a few ways to increase the likelihood that your business stakeholders will prioritize, and your technical/product teams will cooperate with the legal team in, adopting a robust open source software program.

First, partner with your information technology and information security organizations. Rather than focusing solely on the intellectual property-based risks, in some cases, the potential information security risks of noncompliance, such as the consequences of using out-of-date boss components with known vulnerabilities, may be more compelling to your business stakeholders.

As such, your information security team may be a strong co-advocate of a proactive open source software program as a means to keep up to date with known vulnerabilities and available patches.

Second, as much as possible, reduce the burden on the engineers and software developers that will be most directly affected by open source software policies and scans and can often be overlooked by legal teams imposing policies.

The reality is that limiting the software available to your developers — i.e., due to licensing restrictions — and demanding extensive scans and resulting remediation after a build can create a lot of work and distract teams from their ongoing product development efforts.

This can create friction with engineering/developer groups and lead to resistance or delay in aligning on a policy and adoption of a program. Where possible, incorporate automated open source software checks into the software development cycle — e.g., at build time — to raise flags and force conversations among decision makers as the work is being done.

Several software packages are commercially available to integrate with your developers' build tools, manage your established policy decisions on product-by-product bases — after a policy has been developed — and raise any concerns in your engineering project management ticketing systems to drive compliance and remediative action.

Conclusion

The vast majority of commercially available software today includes or is based upon some amount of open source software, which can help developers and engineers more quickly and efficiently create new products. But in-house counsel and compliance organizations should be cautious and measured to monitor and maintain their company's use of such open source software to avoid significant risks and undesirable consequences of doing so.

These are not merely theoretical legal risks but can bring very real business consequences. In fact, some may recall that in the early 2000s, Cisco acquired Linksys for a reported $500 million and soon after learned of unknown code embedded within a flagship Linksys product, the WRT54G wireless router, that was licensed under the copyleft GNU General Public License, or GPL.[1][2]

As a result of the alleged GPL violation, the Free Software Foundation sued Cisco for copyright infringement in 2008.

In 2009 the parties in Free Software Foundation v. Cisco Systems subsequently settled for an undisclosed financial contribution and with Cisco making product source code available.[3]

While the WRT54G wireless router at issue in that dispute has become something of a DIYers' favorite, it is clear that Cisco did not want to make its source code publicly available under the GPL.

To avoid a similar fate, as prudent first steps, companies should incorporate the approaches and tips described here, particularly to establish a well thought out and informed cross-functional policy and routinely proactively audit open source software usage.

While not exhaustive, these will help force productive internal conversations and activities to more effectively mitigate some of the risks of such open source software use to better position companies for long-term success and avoid last minute transaction catastrophes from improper or unknown use of open source software in their products.

Marguerite McConihe is a member and Greg Penoyer is an associate at Mintz Levin Cohn Ferris Glovsky and Popeo PC.

The opinions expressed are those of the author(s) and do not necessarily reflect the views of the firm, its clients, or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.

[1] https://www.linuxinsider.com/story/open-source-and-the-legend-of-linksys-43996.html.

[2] https://tedium.co/2021/01/13/linksys-wrt54g-router-history/.

[3] https://www.pcworld.com/article/529179/article-6776.html.