Blogs
August 30, 2024
Cybersecurity for Lawyers: Open-Source Software Supply Chain Attacks

In this post in our **series on basic cybersecurity concepts for lawyers**, we address open-source software (OSS) supply chain risk.

OSS is software developed using an "open-source" protocol, meaning that its code is fully accessible by the public and often developed by a community of unaffiliated volunteers from around the globe[1] There are many examples of OSS implementations, such as customizable IT solutions that support various "backend" processes such as database management and network monitoring, free software (including operating systems) for devices and users within the organization, and various functions within an orchestration of different pieces of software amounting to some customer-facing application or portal, to name a few. There is a good chance that your organization is using some OSS as a part of its daily operational functions. OSS is ubiquitous and vitally important to many organizations. And while using any software, whether developed and distributed as a proprietary product or developed using an open-source protocol, carries some common risks, there are some distinct differences in how an organization using OSS can address its associated cybersecurity risk.

OSS: An Open-Kitchen Design

Let's say you want to go out for a meal and are hyper-focused on avoiding food poisoning caused by poor kitchen hygiene. You have many options. You can go to a fancy restaurant with a closed kitchen that is very well rated either by other restaurant goers or city health inspectors. Alternatively, you can go to one of the many restaurants with an open-kitchen design where you can watch the kitchen staff at work, allowing you to get some sense as to how the kitchen is run. Even though there is never a guarantee that you can avoid getting sick at a closed-kitchen restaurant, you can still get some sense of the overall risk level without stepping foot inside the kitchen. On the other hand, if you choose an open-kitchen restaurant, whether we are talking about a fast-food sandwich shop or a fancy sushi restaurant, you can see how the food is prepared. Your strategy for preventing food poisoning may be a little bit different at an open-kitchen restaurant. You may focus more on what you see

with your own eyes rather than third-party ratings. But here, too, you cannot completely avoid the risk of a bad night following your meal.

OSS is similar to an open-kitchen design. Whereas the security of proprietary software is backed by market forces and the developer's reputation and contractual guarantees, OSS may not be. But OSS provides transparency in the development process, analogous to an open kitchen. Anyone can inspect the source code and any proposed changes, increasing the chance that error-prone, insecure, or malicious code will be detected and removed or rejected before it is ever incorporated—but only if they are really looking. From a cybersecurity perspective, the fact that the code for software is generally accessible has advantages (e.g., since the code is public, vulnerabilities can be identified more quickly by organizations and others that can seek to remedy them) and disadvantages (e.g., since the code is public, vulnerabilities can be identified more quickly by malicious actors who can seek to exploit them) but is not considered categorically more or less risky to proprietary products.

OSS Supply Chain Attacks

A software supply chain attack is focused on inserting malicious code into, or leveraging an existing vulnerability in, software to gain unauthorized access to the networks and/or devices of the software's users. Supply chain attacks can occur in the context of proprietary software or OSS. However, the best practices and tools available for addressing this risk in the context of OSS differ from addressing that risk with respect to proprietary software. A supply chain attack co-opts the trust in the open-source development model to place malicious code inside the victim's network or computer systems. Essentially, the attacker inserts malicious code, like a foodborne virus, into the software during its development process, positioning the malicious code to be unintentionally installed by the end user installing the software within their network (like an unsuspecting patron eating a sandwich that will ultimately make them sick). Any organization using the affected project has unwittingly invited the malicious code within its walls. Malicious code may already reside within a newly adopted OSS project, or it could be delivered via an updated version of a trusted project. The difference between an OSS supply chain attack and a traditional supply chain attack (e.g., inserting malware into proprietary software) is that the organization using OSS has access to its entire code at the outset and throughout its use (and can therefore examine it for vulnerabilities or otherwise have greater insight into how it functions when used maliciously). While some organizations may have the resources and wherewithal to leverage this as a security advantage, many will not.

Although each project's development community undoubtedly strives to detect and reject malicious contributions, its efforts are not always successful. In 2024, there have been at least two high-profile OSS supply chain attacks: the xz attack and the Polyfill attack. The xz attack, revealed in March 2024, impacted a compression library used by certain types of Linux operating systems.[2] After reportedly working on the project for two years and gaining community trust, the attacker used their governance powers to approve the addition of well-hidden malicious code into a new version of the project.[3] The Polyfill attack's mechanism differed but was also impacting a popular OSS project. Hundreds of thousands of websites using the project reportedly integrated it by importing the code from a particular web domain.[4] A new entity took ownership of that domain in February 2024 and began delivering a version of the project infected with malicious code.[5]

Attacks like these can be difficult to address. Oftentimes, OSS is used to handle less glorious but ubiquitous tasks within a network environment, and even after being made aware of an OSS supply chain attack, it can be difficult to identify a) whether or not your organization actually uses the software in question and b) whether or not any of an organization's service providers use that software as well (this can matter because if the vulnerability or malware allows a malicious actor to breach that provider's network, it can lead to a breach of its customers' networks as well). Of course, those working in cybersecurity in 2021 probably remember the

scramble associated with “Log4j,” a ubiquitous piece of OSS that was found to have a vulnerability allowing a malicious actor to leverage this tool to potentially gain remote access to a network that used the tool.[6] Much of the agita associated with news of this vulnerability had to do with the difficulties in confirming whether it was in use by a particular organization, a critical vendor, or a vendor critical to that vendor.

Addressing the Legal Risks of a Supply Chain Attack

Perfect cybersecurity is impossible, but organizations can take reasonable steps to guard against known categories of threats to help avoid liability after a successful attack.

Here are three questions you should be asking to identify OSS risk:

1. Does your organization have a regularly updated software inventory that identifies any and all OSS used or integrated by your organization?
2. Does your organization have a way of understanding at least its critical vendors’ exposure to OSS risk and how they address it?
3. Does your organization have a policy or protocol for assessing cybersecurity risk associated with its OSS use cases on an ongoing basis (i.e., not only at initial implementation)?

If your organization uses OSS, consider implementing an OSS security policy if there is not one in place. Broadly, a security policy states guidelines and procedures to ensure that the use of OSS is secure. Topics the policy may cover include review of the project’s code, learning of and remediating vulnerabilities, security testing, and regular security training. Implementing such a policy may mitigate the likelihood or impacts of a supply chain attack and otherwise make your organization look much better when the inevitable eventually occurs. And, of course, open-source risk invites open-source solutions, and you can learn more about best practices and other resources from the Open Worldwide Application Security Project (OWASP) Foundation, which is an open-source community dedicated to generating and consolidating some useful resources in this space.[7]

---

[1] OSS is typically offered under one of a handful of standardized open-source licenses. The OSS ecosystem is diverse, and projects vary greatly in complexity, popularity, and governance models. The governance models of many OSS projects permit code contributions from the general public, with a subset of contributors empowered to reject such contributions.

[2] *See* Ionut Arghire, Supply Chain SecuritySupply Chain Attack: Major Linux Distributions Impacted by XZ Utils Backdoor, Security Week (Apr. 1, 2024) https://www.securityweek.com/supply-chain-attack-major-linux-distributions-impacted-by-xz-utils-backdoor/.

[3] *See* Dan Goodin, The XZ Backdoor: Everything You Need to Know, WIRED (Apr. 2, 2024) https://www.wired.com/story/xz-backdoor-everything-you-need-to-know/.

[4] *See* Dan Goodin, 384,000 sites pull code from sketchy code library recently bought by Chinese firm, ars TECHNICA (Jul. 3, 2024) https://arstechnica.com/security/2024/07/384000-sites-link-to-code-library-caught-performing-supply-chain-attack/.

[5] *Id*.

[6] *See* Santiago Torres-Arias, What is Log4j? A cybersecurity expert explains the latest internet vulnerability, how bad it is and what’s at stake, THE CONVERSATION (Dec. 22, 2021) https://theconversation.com/what-is-

log4j-a-cybersecurity-expert-explains-the-latest-internet-vulnerability-how-bad-it-is-and-whats-at-stake-173896.

[7] *See* OWASP, Home Page, https://owasp.org/ (last visited Aug. 23, 2024).

Authors

Explore more in

Privacy & Security Data Security Counseling and Breach Response