How to Manage Open-Source Security and Compliance Risks

31 July 2024 - ID G00803849 - 13 min read

By Analyst(s): Nitish Tyagi, Anne Thomas, Arun Batchu, Aaron Lord

Initiatives:Software Engineering Practices; Build a World-Class Software Engineering Organization

Software development using open-source software fosters innovation but poses numerous security and compliance risks. Software engineering leaders must collaborate with security and legal peers to define OSS management policies and equip their teams with the right tools to implement needed processes.

Overview

Key Findings

- Use of open-source software (OSS) in application development exposes the organization to inherent security and compliance risks as organizations have little to no control over the development of the OSS project. Additionally, the lack of understanding of these risks and actionable policies increases the overall risk.
- Detecting, tracking and mitigating OSS risks in software development is complex, and late discovery can lead to expensive and time-consuming remediation that disrupts the development flow.
- Uncontrolled and unvetted open-source components used in software development can introduce unmaintained, vulnerable, noncompliant and malicious code into the organization.

Recommendations

Software engineering leaders responsible for secure and efficient software development should:

- Mitigate OSS risks by collaborating with security and legal peers to co-create actionable, comprehensive, and pragmatic risk mitigation policies and practices.

- Uncover OSS risks as early as possible by choosing an appropriate software composition analysis (SCA) toolkit, and implementing automated scanning in the DevOps pipeline.
- Control what open-source code can be used by your developers by establishing an OSS repository managed either by an internal team or a third-party subscribed service.

Strategic Planning Assumption

By 2027, 80% of software engineering teams will implement SCA tools in their workflow, up from 50% today, to minimize the security and licensing risks associated with OSS.

Introduction

Modern software solutions heavily rely on OSS, with more than 95% of organizations increasing or maintaining their use of OSS. [1] On the flip side, software supply chain attacks are becoming increasingly sophisticated (see How Software Engineering Leaders Can Mitigate Software Supply Chain Security Risks and Leader's Guide to Software Supply Chain Security), while OSS intellectual property risk management is getting especially tricky, with more than 2,000 OSS license variants in existence today. According to the latest survey results, the number of risks has almost tripled over the last year:

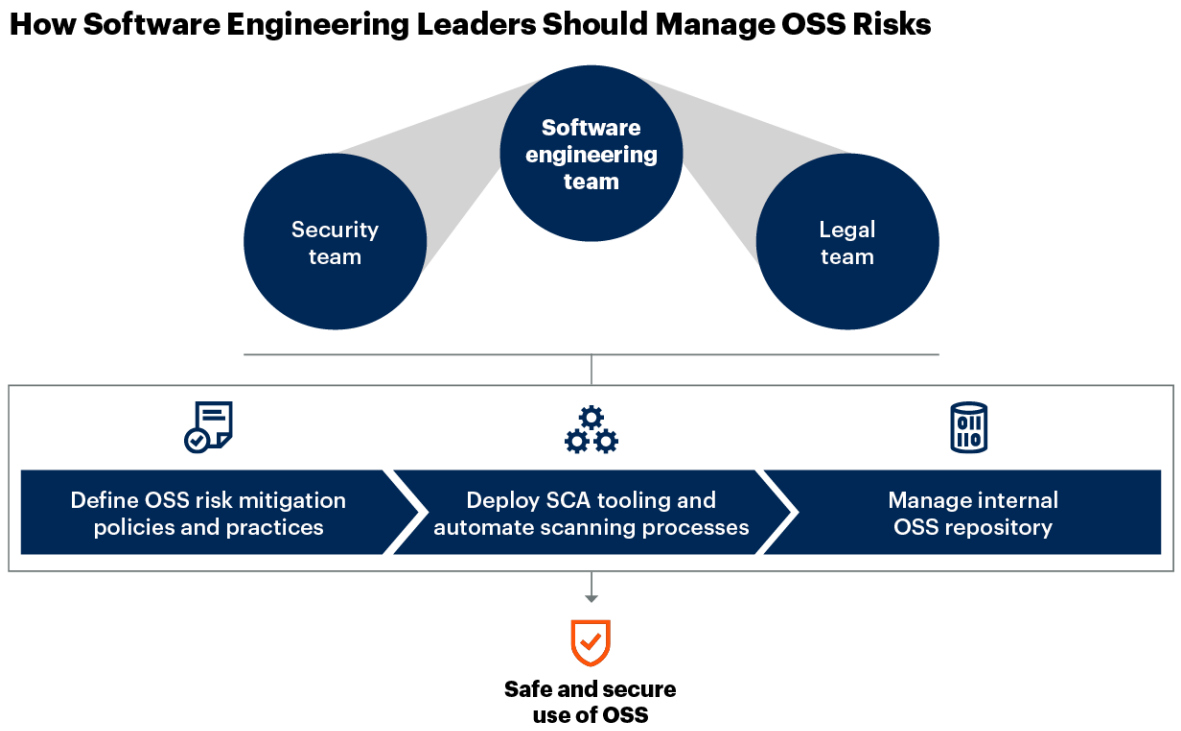
- 90% of the code running in production is of open-source origin. Disturbingly, one in eight open-source downloads includes a known risk. [2]
- 96% of the production codebases examined by the Synopsys analysis tool contain open-source code, and 84% contain known risks and vulnerabilities. [3]

> **The number of malicious open-source packages increased from 88K in 2022 to 245K (280%) in 2023.**
>
> *— 9th Annual State of the Software Supply Chain, Sonatype2*

The benefits of using OSS outweigh the risks, but how can software engineering leaders protect their enterprise from the inherent risks associated with license violations, security vulnerabilities and supply chain attacks?

Seek support from your security and legal peers, and use this research to ensure safe and secure use of OSS by developing OSS risk mitigation policies, automating scans with tools like SCA and maintaining an internal OSS repository (see Figure 1):

Figure 1: How Software Engineering Leaders Should Manage OSS Risks



Source: Gartner
OSS = Open source software; SCA = Software composition analysis
803849_C

Gartner

Analysis

Define Risk Mitigation Policies and Practices

Using OSS brings a number of benefits in the form of innovation, accelerated software development and cost savings; however, it does introduce inherent security and compliance risks. Additionally, undermanagement of OSS significantly increases the potentiality of these risks. Software engineering leaders must ensure that the following questions are answered:

- Who is responsible for tracking vulnerabilities in OSS components?
- How do we track which software products are potentially impacted by those vulnerabilities?

- How do we ensure that patches are applied to all vulnerable software?
- How do we ensure that our software products use the latest patched versions of the OSS components?
- How do we ensure that we don't violate the OSS licensing terms?
- How do we prioritize which issues to address first, and which ones can be deferred, and for how long? What class of issues should stop the release pipeline?

Your first step should be to work with your software engineering organization to create risk mitigation policies and practices for using OSS safely and efficiently. Don't expect software engineers to have expertise in understanding detailed OSS licensing terms or what's required to prevent or mitigate severe vulnerabilities from entering production systems. Software engineering leaders must collaborate with their peers in the legal and security management organizations to define OSS risk mitigation policies and practices.

According to the 2023 Gartner Security in Software Engineering Survey, improving collaboration between software engineering and security staff can lead to up to a 27% improvement in software security outcomes. However, only 29% of respondents agreed that both teams collaborate effectively (see Infographic: 3 Ways to Improve Software Engineering & Security Collaboration). [4] You should also enlist the support of legal peers to develop risk policies against the complex terms and conditions of open-source licenses. OSS risk management for both security and legal should be included in your overall OSS governance policy.

An OSS governance policy defines the organization's OSS charter, purpose, rules, guidelines, responsibilities and authorities for all aspects of use and contributions to OSS efforts (see How to Create and Enforce a Governance Policy for Open-Source Software). A critical section of the OSS governance policy defines OSS management and risk mitigation requirements that answers the above questions. Organizations with a mature OSS practice typically set up an open-source program office (OSPO) to define and own the OSS governance policy. The OSPO (or a subcommittee) also defines the OSS management policies, procedures, standards, guidelines, processes and practices that govern the day-to-day usage of OSS. If you don't have a formal OSPO, you should create an OSS governance committee to define the ownership and accountability for creating, managing, reviewing and updating the governance policy. These lines of ownership can be defined via responsible, accountable, consulted and informed (RACI) charts.

OSPOs (or the OSS management subcommittee) should review OSS management policies and procedures at least annually and retrospect the policies for root cause analysis after any breach or incident. OSPOs should adopt a continuous delivery model for improving governance automation of the processes and practices that ensure compliance (see Best Practices for Setting Up an Open-Source Program Office).

Once the OSS governance policy is formulated, problem notifications and remediation workflows among the three teams (software engineering, security and legal) need to be automated to resolve issues as quickly as possible. Additionally, you must ensure that the SCA tool you choose has workflow features that fit your organizational needs.

Establish clear capacity commitments and service levels (such as time to respond) you expect from your security and legal teams. Unfortunately, the growing disparity in the ratio of developers to security and legal experts is creating capacity challenges. This means there is a pressing need to support and augment the capabilities of your security and legal teams to meet these expectations. One step is to build application security skills in your software engineering teams.

According to the 2024 Gartner Software Engineering Survey, three out of four software engineering leaders consider application security as a highly important skill to deliver software that meets the business needs of their firm. [5] Moreover, the 2023 Gartner Security in Software Engineering Survey states that more active participation of software engineering teams in security tasks leads to better software security results (see Case Study: Building Security Skills in Software Engineering Teams (Siemens Healthineers). [4]

Deploy SCA Tooling and Automate Scanning Processes

Tools are a vital investment to facilitate OSS management, and the foundation for your OSS management tool suite is the SCA. These application security testing tools analyze code and related artifacts (containers, registries, etc.) to detect open-source and third-party software components known to have security and functional vulnerabilities. They also identify out-of-date components for security patches, or those that pose licensing risks. SCA products and services help ensure that the software supply chain includes only components that adhere to your policies and, therefore, supports secure application development and assembly. The tools can support the following capabilities:

- Generate a software bill of materials (SBOM) that tracks the software supply chain. SBOM is formally structured, machine-readable metadata that uniquely identifies a software package and the components used to build it. The components can be open source or proprietary. Creating, safeguarding and maintaining SBOMs is also a recommended practice under the Secure Software Development Framework by NIST. [6]
- Identify any known vulnerabilities from the common vulnerabilities and exposures (CVEs) database. Some tools maintain their own database as well.
- Generate a vulnerability exploitability exchange (VEX) document that provides more information about the vulnerability, and determines whether the vulnerability is exploitable in the context of the software product. VEX helps in analyzing the reachability of known vulnerabilities.
- Track and identify the vulnerable direct and transitive dependencies throughout the open-source pipeline.
- Send remediation recommendations to developers. Some tools also provide a comprehensive list of required changes in your application once you upgrade to the secured version.
- Update affected teams on availability of a security patch or updated version of the OSS dependency.

Software engineering leaders should seek help from security peers while selecting the SCA tool. For more information on SCA and related application security testing technologies, see:

- Magic Quadrant for Application Security Testing
- Mitigate Enterprise Software Supply Chain Security Risks
- Emerging Tech: A Software Bill of Materials Is Critical to Software Supply Chain Management
- Innovation Insight for SBOMs

Representative vendors in the SCA space: Anchore, Backslash, Black Duck by Synopsys, Checkmarx, Endor Labs, Insignary, Mend.io, Snyk and Sonatype.

Automate Scanning Processes

Software engineering leaders should ensure that SCA and other software security tools are built into the automated DevOps pipeline. Triaging vulnerabilities and licensing issues, and patching application libraries manually, can add enormous delays and consume a tremendous amount of scarce enterprise resources in terms of both time and energy (see Trends to Guide Security Automation Decisions in Software Engineering).

SCA tools automate the scanning of application dependencies, and identify vulnerabilities and licensing issues much faster than human experts can. They prioritize the issues based on vulnerability reachability, predefined policies and can even generate pull requests to automatically fix the problems.

According to the 2023 Gartner Security in Software Engineering Survey, automating security tasks can bring up to 1.15x better security results. [4]

Software engineering leaders should:

- Ensure that the SCA tool detects problems as early in the development cycle as possible by adopting SCA connectors that plug into their integrated development environment (IDE). Some SCA tools provide the capability to add the governance rules, so that SCA scans adhere to the governance policy.
- Ensure that the SCA tools are integrated into the workflow to scan immediately when code is committed, and to immediately raise any potential security and legal risks.
- Ensure every release is scanned. Prohibit deployment if any issues exceed established severity levels or belong to a nonapproved license category, as specified in the governing policies.

Manage an Internal OSS Repository

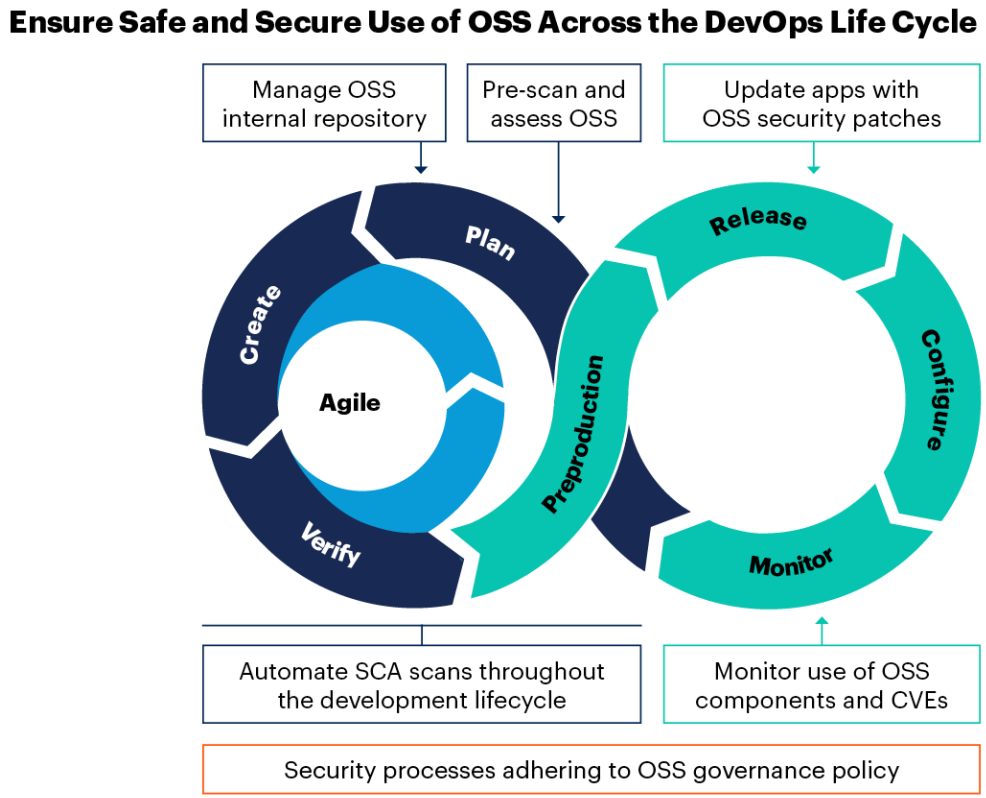
A recent attack on GitHub impacted more than 100,000 repositories, resulting in developers using the malicious code instead of the original one. [7] To prevent such attacks, software engineering leaders should restrict software engineering teams from directly downloading or installing open-source code into their application from the internet. Instead, all the required OSS components should be hosted and managed on an internal repository. This is also a recommended practice under the Secure Software Development Framework by NIST. [6]

This internal repository can either be maintained by an internal team (with support of the platform engineering team) or by a third-party service provider. When using a third-party service provider, your OSS governance teams should work with them to ensure that OSS is being managed according to the company's OSS governance guidelines.

When creating the internal repository, each approved OSS component should be scanned for vulnerabilities or license risks using your preferred SCA tool. OSS projects should be assessed for governance compliance, project health, and quality and security practices prior to inclusion in the repository (see 3 Steps for Assessing an Open-Source Software Project). A few vendors such as Endor Labs and Tidelift support OSS assessment and OSS repository management services. Artifact repository management services such as JFrog Artifactory and Sonatype Repository Firewall provide automated scanning capabilities to identify known CVEs and legal risks. Organizations can use artifact repository management services with any assessment tool to get the desired results.

Figure 2 illustrates the use of tools and practices across the DevOps life cycle for managing OSS effectively.

Figure 2: Ensure Safe and Secure Use of OSS Across the DevOps Life Cycle



Source: Gartner
OSS = Open source software, SCA = Software composition analysis, CVE = Common vulnerabilities and exposures
803849_C

Gartner

To effectively manage OSS, software engineering leaders should:

- Adopt the shift-left approach, and prescan and assess the requested OSS components.
- Host the approved OSS components on the managed internal OSS repository.
- Integrate SCA to your development environment, and automate scanning of code for unmanaged OSS components or new vulnerabilities. Triage and remediate the vulnerabilities with the help of your SCA tool.
- Ensure that the deployed application is using upgraded security patched versions of OSS components in cases of known vulnerabilities.
- Monitor the use of OSS components, and subscribe to the notifications from relevant sources such as mailing lists or security advisories to get instant information about new vulnerabilities, and available security patches and product releases.

Evidence

[1] 2024 State of Open Source Report, OpenLogic.

[2] 9th Annual State of the Software Supply Chain, Sonatype.

[3] 2024 Open Source Security and Risk Analysis Report, Synopsys.

[4] **2023 Gartner Security in Software Engineering Survey.** This survey was conducted online from 7 June through 14 July 2023. It sought to understand the different aspects of security practices, such as responsibilities, skills, metrics, requirements, processes, tools and technologies, and security roles in software engineering. In total, 300 software engineers and security professionals, up to the role of senior vice president, across industries participated. The respondents were from North America (n = 178), EMEA (n = 76) and Asia/Pacific (n = 46). *Disclaimer: The results of this survey do not represent global findings or the market as a whole, but reflect the sentiments of the respondents and companies surveyed*.

[5] **2024 Gartner Software Engineering Survey.** This survey was conducted to identify the most important roles and skills for software engineering leaders and the change in their demand and importance since last year, understand how talent is sourced generally and for acquiring necessary artificial intelligence (AI)/machine learning (ML) skills, and what tools are seen to increase developer productivity and the metrics used to measure them. It also examines how software engineering leaders anticipate change in their operating budgets and the cost management steps taken. It further aims to identify the quality and testing techniques and programming languages software engineering leaders currently use and/or plan to use; their frequency of usage of user experience (UX) design, user research and AI in generating components of user experience; and its impact on user satisfaction, accessibility and usability. It also intends to understand the software engineering leaders' responsibilities they find most difficult, the career paths available for senior-level individual contributors and how they are set up, how organizations attract and retain top performers in those career paths, and what management training is offered to staff. The survey was conducted online from October through December 2023 among 300 respondents from the U.S. (n = 241) and the U.K. (n = 59). Qualifying organizations operated in multiple industries (excluding the IT software industry and education sector) and reported enterprisewide revenue of at least $250 million or equivalent for fiscal year 2022, with 63% over $1 billion in revenue. Qualified participants were highly involved in managing software engineering/application development teams and the activities they perform. *Disclaimer: The results of this study do not represent global findings or the market as a whole, but reflect the sentiments of the respondents and companies surveyed*.

[6] Secure Software Development Framework (SSDF) Version 1.1, NIST.

[7] GitHub Besieged by Millions of Malicious Repositories in Ongoing Attack, Ars Technica.

Document Revision History

How to Manage Open-Source Software Risks Using Software Composition Analysis - 27 November 2020

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

How to Create and Enforce a Governance Policy for Open-Source Software

Tool: OSS Governance Policy Template

Quick Answer: Choosing the Ideal Support Strategy for Open-Source Software

Mitigate Enterprise Software Supply Chain Security Risks

Structure Application Security Tools and Practices for DevSecOps

Trends to Guide Security Automation Decisions in Software Engineering

---