ANALYST BRIEF
Sponsored by: Cycode

**Software composition analysis (SCA) solutions cannot detect or stop many software supply chain attacks because they look at only open source dependencies. Organizations need to account for dependencies across the entire software development life cycle (SDLC) to protect their software supply chains.**

# Software Supply Chain Security Is More Than Just Tracking Open Source

*April 2023*

**Written by:** Jim Mercer, Research Vice President, DevOps and DevSecOps

## Introduction

The recent pandemic accelerated the growth of internet-exposed applications, including the rising use of open source, containers, and APIs. IDC forecasts that there will be 750 million new applications by 2025 (*750 Million New Logical Applications: More Background,* December 2021). Unfortunately, the exponential growth of applications and digital commerce has met with unprecedented security breaches. While applications can support critical business success, the software supply chains that build them become top targets for cyberattacks.

The software supply chain comprises everything and everyone that touches an organization's application in the software development life cycle (SDLC), including people, processes, dependencies, and tools. The security of the software supply chain has been overlooked, and bad actors now see the software supply chain as a soft target. Attackers are becoming more sophisticated in hiding from detection, growing more patient and subtle, and taking time to learn about the environment before attacking.

Organizations are still unclear about what aspects of the software supply chain to secure and how to secure those aspects. A recent IDC survey found that organizations are less confident in their software supply chain security compared with other aspects of their DevSecOps capabilities (*DevSecOps Adoption, Techniques, and Tools Survey, 2022,* IDC#US48599822). Familiar application security testing solutions (i.e., SAST, DAST) focus on application code security but not the security of the supply chain used to build the application.

## AT A GLANCE

### KEY TAKEAWAYS

- The growing threat to the security of the software supply chain led to the creation of U.S. Executive Order 14028.
- The software supply chain comprises everything in the software development life cycle (SDLC).
- Software supply chain security must include more than software composition analysis (SCA), which looks at only open source.
- Organizations need full SDLC software supply chain security.
- An SDLC approach to software supply chain security provides an analysis of vulnerabilities across the entire software delivery pipeline.

## SCA Is Only Part of the Solution

Much of the focus on software supply chain security has been on open source software (OSS). Using OSS components helps teams build applications with greater velocity, but it also puts organizations at risk of deploying with security vulnerabilities that may be included as part of the OSS. When a single OSS component is adopted in multiple applications across different companies, the blast radius of an OSS vulnerability grows, making OSS an attractive target.

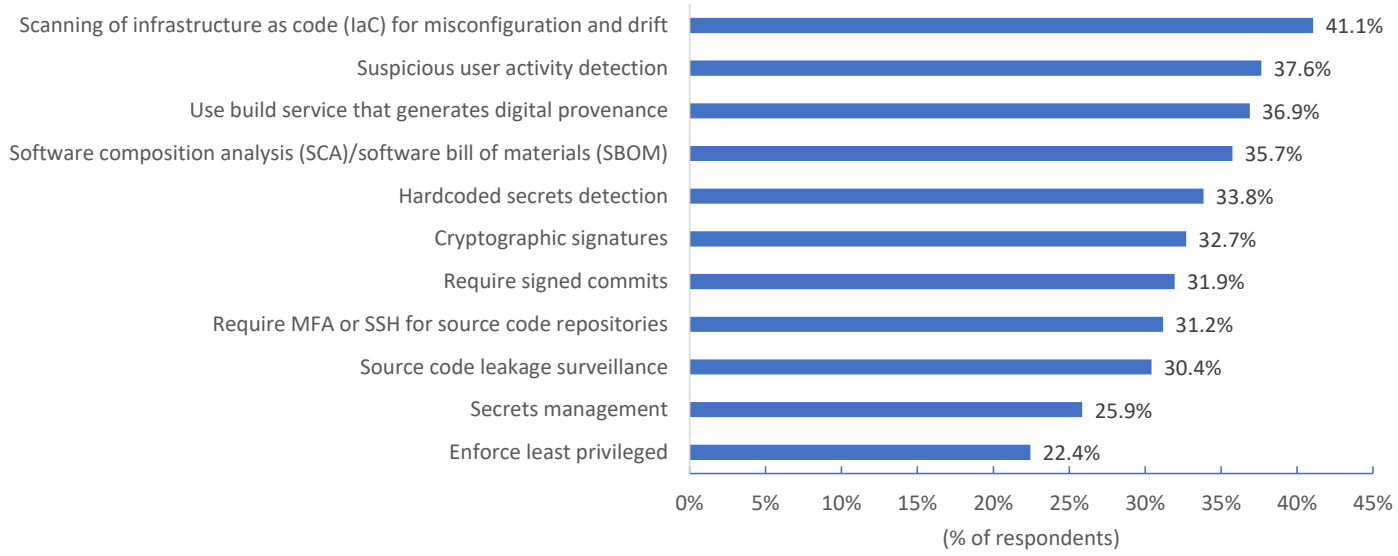
To address these concerns, organizations often turn to software composition analysis (SCA), which identifies potential risks and vulnerabilities of OSS artifacts integrated into an application by comparing dependencies, signatures, or binaries against a database of known threats and vulnerabilities. SCA is an important part of software supply chain security. However, many organizations now mistakenly consider SCA and the resulting software bill of materials (SBOM) as their approach to software supply chain security. This approach ignores development pipelines, often the entry point in software supply chain attacks.

## SCA Is Limited in Scope

Unfortunately, SCA provides a limited picture of software supply chain security risks and does not encompass the complete SDLC. To get a full picture, organizations must incorporate all the dependencies across their software delivery pipelines, including continuous integration/continuous delivery (CI/CD), infrastructure as code (IaC), and other development and DevOps tools. Figure 1 illustrates how SCA is only part of an overall software supply chain security approach.

FIGURE 1: ***Different Practices Used for Securing the Software Supply Chain***

***Q What practices have you implemented for software supply chain security in your organization [select all that apply]?***




*n = 263*

*Source: IDC's DevSecOps Survey, May 2022*

SCA solutions cannot detect or stop well-known software supply chain attacks such as SolarWinds, where attackers presumably created a back door via the TeamCity CI/CD solution, or Codecov, where a code coverage solution allowed an attacker to upload an altered bash script to steal credentials and infiltrate downstream customers. As such, the software supply chain is broader than just OSS dependencies, and standard SCA solutions fall short of securing all of an organization's software supply chain dependencies.

## The Need to Consider an SDLC Approach

Various other non-OSS dependencies and attack vectors in the supply chain need consideration across the software delivery pipeline. These can include an organization's code, third-party dependencies in DevOps tools, plug-ins, automation scripts, infrastructure, registries, artifact repositories, and the developers working in the SDLC.
A dependency is essentially anything an organization uses to create an application that the organization doesn't control that could be vulnerable and impact the software's quality and security.
Organizations need to examine both the dependencies in code and the dependencies within their software delivery pipelines. Because bad actors often find the weak link in the supply chain and can move laterally across the pipeline, they are able to hide their exploits and do more damage by taking advantage of DevOps automation.

An SDLC approach to software supply chain security provides the end-to-end visibility needed to defend against complex threats that span different aspects of the software delivery pipeline, akin to a home security system that looks at all possible home entry points (doors, windows, etc.), not just one.

Organizations must take a full SDLC approach to software pipeline security to reduce exposure to supply chain attacks. The software supply chain must be protected at all points of entry — one cannot simply lock the front door and leave the back door open. Thus organizations must uncover all exposures concerning dependencies across the delivery pipeline to identify openings for sophisticated covert attacks. It is possible for an organization to protect its software supply chain only when it has visibility into the security of all pipeline dependencies.

## Benefits

Organizations can benefit from a deeper understanding of the entire software delivery pipeline that encompasses the whole SDLC instead of just an SCA solution. A complete platform provides an in-depth analysis of vulnerabilities across the entire pipeline, including OSS and other dependencies such as development tools and build modules, which have their own dependencies. Additional benefits include:

- Faster and more accurate identification of vulnerabilities across code and deployments
- Improved prioritization of vulnerabilities that considers both reachability and exploitability
- Analysis of the tools used across the SDLC to identify whether they contain any known vulnerabilities
- Assurance of security best practices for the development tools across the software delivery pipeline to ensure they are not configured in a way that can create a security risk

However, the most compelling benefit is the reduced risk to the organization by having insights into the entire risk exposure across the SDLC.

### Visibility

An SDLC approach to application security and software supply chain security provides the end-to-end visibility needed to defend against complex threats that span different aspects of the software delivery pipeline. This is akin to a home security system that looks at all possible home entry points (doors, windows, etc.), not just one. As such, organizations must ingest and examine data across the different aspects of the software supply chain, tracing activity from code to

software deployment. Understanding what makes up the software delivery pipeline then enables organizations to identify vulnerabilities and puts them in a position to lock down, harden, or mitigate any pipeline vulnerabilities. Some potential software delivery pipeline vulnerabilities organizations need to consider are:

- Source code
  - Third-party OSS
- Development tool exposures
  - Embedded OSS vulnerabilities
  - Tool-level vulnerabilities (i.e., NPM, Jira, Codecov breaches)
- Build modules (i.e., GitHub Actions)
  - Can contain OSS and/or be tampered with
- IaC security and tools
  - An unpatched vulnerability in an IaC tool could serve as a threat entry point to the core infrastructure
  - Insecure configurations
- Production configuration drift
  - Configurations falling out of sync with their expected secure state
- Insecure cloud configurations
  - Gaps or errors that could expose an organization's application to risk when deployed to the cloud
- Runtime environment exposures
  - Exposed CI/CD secrets
  - Insecure tool configurations

### *Coherent Understanding of Risk*

Most common vulnerabilities and exposures (CVEs) are published with a Common Vulnerability Scoring System (CVSS) score of high or critical, so every vulnerability appears to be an emergency. Some SCA solutions have some intelligence to understand whether the vulnerabilities are code reachable within application code, reducing false positives that waste developers' time. However, using an SDLC approach with end-to-end insights, organizations can better understand risk across the SDLC beyond generic CVSS vulnerabilities flagged via standard SCA solutions.

An SDLC approach considers all the potential dependencies that may be vulnerable across an organization's software delivery pipelines. Because data from across the entire pipeline is ingested, an organization can ascertain if the vulnerability is exploitable based on the organization's application/cloud configuration and environments. The location of vulnerable dependencies in code and production deployments can be pinpointed, allowing organizations to track and quickly patch vulnerabilities based on the analysis of the entire SDLC. Further, streamlining the remediation of vulnerabilities enables development resources to focus on areas of greatest risk first.

## Considerations

Organizations should consider emerging software supply chain security platforms that provide the most comprehensive insights and intelligence on the security dependencies of the entire SDLC.

A "build your own" SDLC software supply chain solution would be a momentous undertaking that is likely cost prohibitive and impossible for most organizations to take on. It requires the procuring and deploying of multiple point security solutions, from which siloed data from disparate tools must be ingested, normalized, and analyzed to identify the correct security signals.

## Key Trends

The threat to software delivery pipelines continues to expand such that Royal Hansen, Google's vice president of engineering for privacy, safety, and security, recently blogged that software supply chain security is among "the most critical national security risks facing governments worldwide."

The increase in attacks on the software supply chain is compelling the U.S. federal government to use its purchasing power to raise security standards through actions such as the 2021 U.S. Executive Order 14028 on Improving the Nation's Cybersecurity. On February 4, 2022, the National Institute of Standards and Technology (NIST) issued guidance via two documents: *Secure Software Development Framework* and *Software Supply Chain Security Guidance*.

As more organizations adopt these standards, nongovernmental organizations purchasing software will expect the same due diligence from their business partners or vendors. IDC predicts that securing the software supply chain will be a core competency that 75% of large digital innovators will embrace by 2023 (*IDC FutureScape: Worldwide Future of Digital Innovation 2022 Predictions*, October 2021).

## Conclusion

The number and complexity of software supply chain attacks continue to escalate. As a result, every organization building software is now a target for a software supply chain attack and thus must be diligent to avoid becoming the next victim of a high-profile security breach. Securing OSS components via SCA is important but does not provide a complete supply chain solution. Organizations must take an SDLC approach that looks at all the software delivery pipeline dependencies and potential vulnerabilities.

Collecting and analyzing data across modern software delivery pipelines is complex and not something most organizations can undertake independently. Therefore, organizations should consider emerging platforms that provide comprehensive insights and intelligence on the security of dependencies across the entire SDLC.

# About the Analyst



***Jim Mercer,*** *Research Vice President, DevOps and DevSecOps*

Jim Mercer is a Research Vice President within IDC's DevOps and DevSecOps Solutions research practices. In this role, he is responsible for researching, writing, and advising clients on the fast-evolving DevOps and DevSecOps markets. Mr. Mercer's core research includes topics such as rapid enterprise application development, modern microservice-based packaging, GitOps, application security, software supply chain security, and automated deployment and life-cycle/management strategies as applied to a DevOps practice.

## MESSAGE FROM THE SPONSOR

**About Cycode**

Cycode's modern approach to application security enables organizations to effectively secure their cloud-native applications with cost-efficient use of tooling and staff across the SDLC. Cycode provides:

- A single unified software supply chain security platform that consolidates SAST, SCA, container scanning, IaC scanning, pipeline security and governance, code tampering, secret scanning, and code leak detection
- Complete visibility of application security risks with rapid deployment and agentless integrations
- Alert prioritization that optimizes remediation based on event correlation from code to cloud
- Developer workflow integrations that incorporate security processes into development while giving security teams the visibility and control to reduce risk

Cycode gives security teams the ability to drive digital transformation through a unified, cost-effective program that delivers:

- High-efficiency risk mitigation
- Tool consolidation and cost reduction
- Operational efficiency and business acceleration

To learn more, visit www.cycode.com.

IDC Custom Solutions

The content in this paper was adapted from existing IDC research published on www.idc.com.

**IDC Research, Inc.**
140 Kendrick Street
Building B
Needham, MA 02494, USA
T 508.872.8200
F 508.935.4015
Twitter @IDC
idc-insights-community.com
www.idc.com

This publication was produced by IDC Custom Solutions. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2023 IDC. Reproduction without written permission is completely forbidden.