

The Foundational Role of Security and Governance in the Software Supply Chain

By Samir Sakpal, Frost & Sullivan



A Frost & Sullivan White Paper

Powering clients to a future shaped by growth

F R O S T  S U L L I V A N

CONTENTS

Executive Summary	3
Why Thinking Through Software Supply Chain Attack Security Is Increasingly Important.	4
Governance and Security Policy: Differentiating Value Proposition	6
Recommendations and Call to Action	7
Cycode: Pioneering Solution for Software Development Pipeline Security.	8

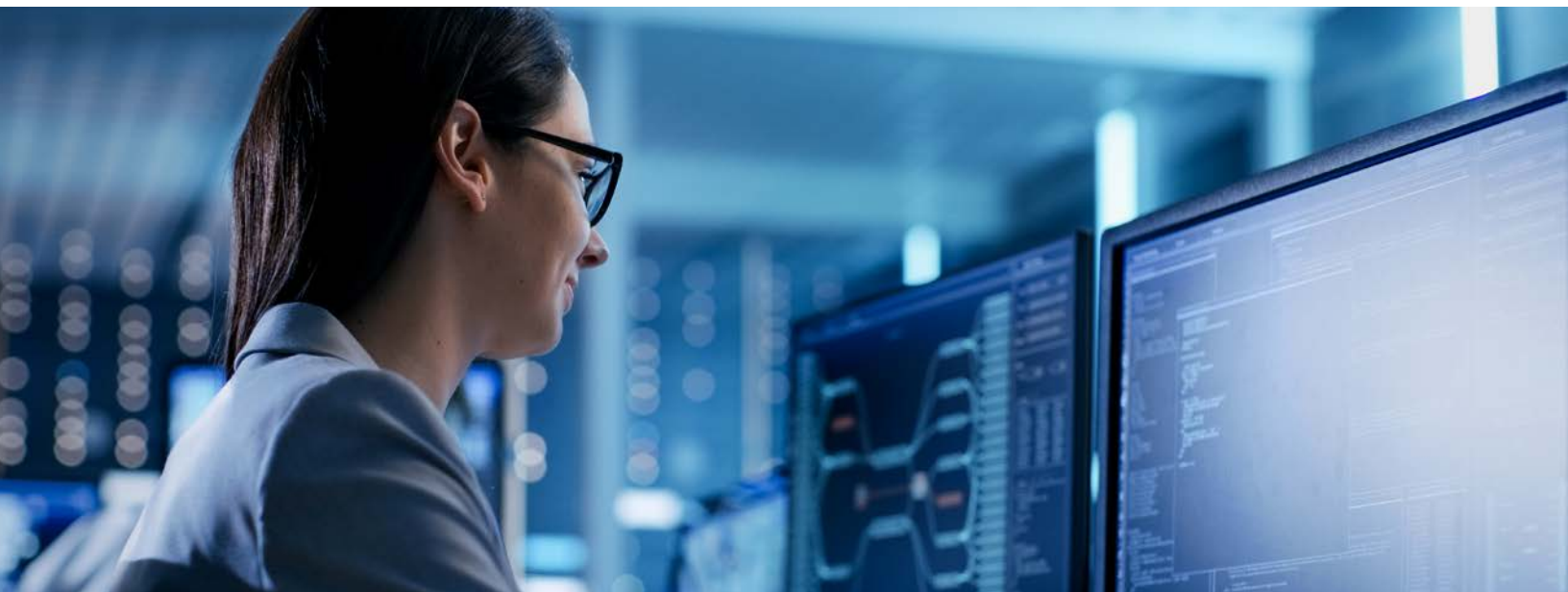
Executive Summary

Modern Software Development Demands Security Attention

Enterprises of all sizes and across all industry verticals have been evolving their IT ecosystems through rigorous and continual software development. The result is increasing prominence of a modern software development pipelines that—from early phases through production—include DevOps tools and infrastructure. Unfortunately, security teams have minimal visibility into DevOps tools and infrastructure, and the visibility that does exist is generally limited to one specific software delivery life cycle (SDLC) phase or environment. This limited visibility makes it difficult for security teams to understand what activities are taking place, what security controls exist, and how consistently the controls are applied across the many DevOps tools and SDLC phases.

Moreover, intruders may use DevOps tooling as an attack vector. Many organizations purchase, implement, and maintain DevOps tools without involving their security teams. Because these tools house mission-critical data such as source code and reside outside of a security team's purview, they are an attractive and vulnerable target for attackers looking to compromise software development.

As enterprises invest in DevOps and continue increasing the agility of their software and application development processes, the need intensifies to secure the emerging software supply chain. The ability to understand actions taken within and to implement consistent security and governance policies across modern DevOps infrastructure is an easily overlooked yet integral part of any cybersecurity strategy—one that greatly reduces the likelihood of an organization experiencing a software supply chain attack.



Why Thinking Through Software Supply Chain Attack Security Is Increasingly Important

Securing modern software supply chains from attack is not an easy task. Several factors of modern software development processes make the software supply chain inherently difficult to secure, including the following four aspects.

Diverse Tooling with Inconsistent Settings

Organizations often have multiple development teams. Each team implements the tools and infrastructure that are most suited to help it effectively and efficiently build its applications. This means different teams may have different tools, and with acquisitions and mergers, even more diverse tooling is added to the DevOps infrastructure. By default, teams configure the tools with efficiency—not security—in mind, and thus they require adjustments to align with security best practices to reduce risk. Security policies and settings are different across the myriad of tools available; sometimes, such as with SCMs, settings must be configured on a repo-by-repo basis. This wide scope of configuration possibilities creates an enormous barrier to implementing consistent security.

Insider Threats in the SDLC

Enterprises are increasingly dependent on in-house software development using their own teams to build applications designed to interact with customers or to sell as a source of revenue. In many ways, this practice has escalated the risk of security attacks, not just from outside but also from within the organization, in the form of both compromised developer accounts and malicious insiders. Developer accounts are attractive targets for attackers because they traditionally include overprovisioned access under the assumption that developers require these privileges to perform their day-to-day activities.

By default, teams configure the tools with efficiency—not security—in mind, and thus they require adjustments to align with security best practices to reduce risk.

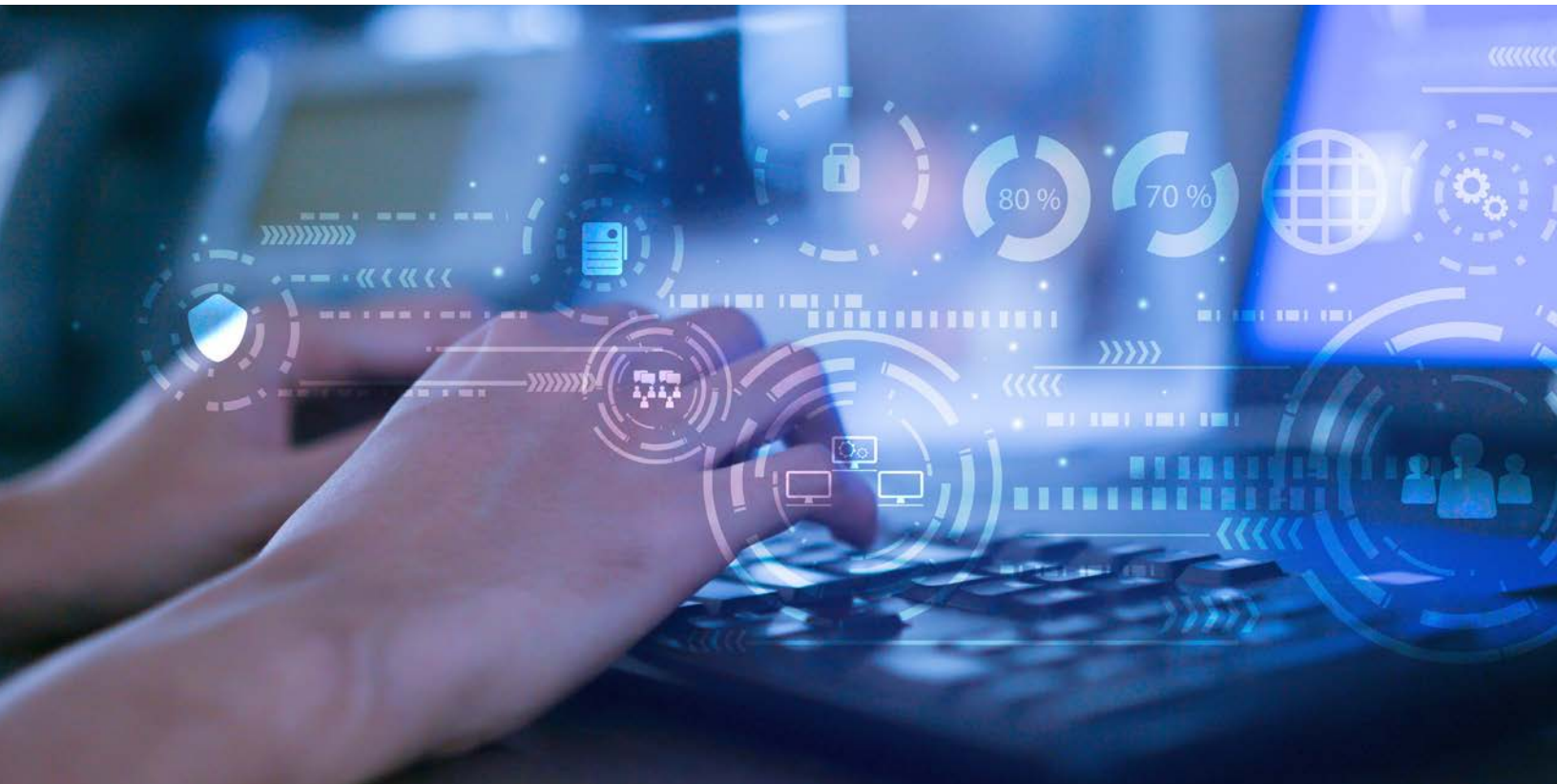


The Ease of Lateral Movement

Regardless of whether attackers compromise a supply chain by way of credential theft, code leakage, vulnerability, or some other means, the same automation and interconnectedness inherent to the DevOps movement make it easier for attackers to move laterally. For example, the everything-as-code and GitOps approaches establish trusted pipelines between the development and production environments. An attacker might use a compromised developer account to insert and sign malicious code into a code repository. Once signed, code is committed into the repository; automated processes launch a build using that code, perform or bypass testing and scanning as defined in the commit, and even deploy it into production. In other words, attackers that have the ability to sign and commit code also have the ability to access the production environment.

The Lack of Appropriate Tooling

Traditionally, the application security tooling available to most organizations has focused solely on securing either source code during development or in production applications. However, a critical gap exists in tooling that, if addressed, can help organizations efficiently and effectively secure the entire tooling pipeline. As a result of this gap, software delivery pipelines often remain vulnerable to attacks that might otherwise be prevented if security postures were hardened.



Governance and Security Policy: Differentiating Value Proposition

Many modern organizations' ability to generate revenue and interact with their customers is dependent on delivering high-quality software. While quickly developing applications that have cutting-edge features and functionality can provide a competitive advantage, the mission-critical nature of modern software also means defects and security flaws can have larger ramifications than they did in the past. Thus, organizations must balance efficiency with security: both are directly tied to success.

Enterprises must also focus on standardizing security and governance policies to drive consistency across the entire DevOps pipeline. A hand-in-glove working practice between the development and security teams is critical. Once established and validated, this approach allows organizations to include security and governance as part of their larger project and IT strategy across the entire software supply chain. An overall governance strategy helps ensure developers have access only to the code they need, which minimizes the risk of a developer compromise affecting software integrity.

The mission-critical nature of modern software also means defects and security flaws can have larger ramifications than they did in the past.



Continuous governance and security policy enforcement are imperative for maintaining application health and organizational integrity. The need for comprehensive monitoring, deep visibility, policy settings, and real-time resolutions to issues is irrefutable. Organizations need an end-to-end, comprehensive security platform for their software development supply chain that effectively and efficiently mitigates all threats and risks.

Implementing consistent and effective security policies throughout software delivery pipelines also reduces the risk of attacks that result in breaches. For example, hardcoded secrets become dangerous when attackers gain access to them. The most common ways for secret exposure to happen include compromised insiders, malicious insiders, and code leakage. Policies like strong authentication reduce attackers' ability to compromise relevant developer accounts. Least privilege policies make it difficult for would-be attackers to infiltrate because they must have access to the correct account and privileges. Moreover, least privilege policies reduce the pool of developers who could accidentally leak code to public code-sharing sites or repos. In all of these cases, implementing security policies reduces the risk of a software supply chain breach. The most salient takeaway here is that proper security and governance instills security best practices that improve the efficacy of all other SDLC security projects.

Recommendations and Call to Action

► Enforcement of Least Privilege Policies and Access Control across the Software Supply Chain

Managing security policies through a central security platform is an essential measure to harden tooling across an enterprise's software development pipeline. Security teams must implement tooling and processes to continuously monitor and revise access and privileges for system users. This includes detecting excess privileges and stale users, deprovisioning departed employees, and enforcing separation of duties. The best way to perform these actions is through an easy-to-manage security platform that integrates all of the various elements in the pipeline.

► Hardening Authentication for DevOps Resources

In continuous integration/continuous delivery (CI/CD) software development environments, developers are constantly working on developing, reviewing, and revising applications. This practice requires the provision of multi-level access to the development pipeline as well as the databases that contain sensitive data outputs from the software development process. Enterprises need a security platform that ensures strong authentication policies are in place across all tools in use to confirm that developers are who they say they are.

► Implementing Branch Protection Rules and Build Rules

It is imperative for enterprise IT to ensure security is baked into their software supply chain through multiple mechanisms. As part of securing the software supply chain across every component and element, setting effective branch protection and security-oriented build rules is essential. Branch protection rules help to keep a particular branch in a safe state. For example, requiring peer review and disabling forced push and delete operations on an organization's main branch helps ensure that any changes to the branch have gone through multiple levels of review. Establishing build rules can prevent tampering, for example, by ensuring that build inputs come only from secure sources, including the verified output of previous build steps, and that provenance is documented accurately.

► Monitoring Change Control to Ensure Security Controls Are Not Turned Off

Today's software development pipelines include a host of security controls, both inside the pipeline (e.g., application security testing and software composition analysis) and, as this paper discusses, along the overall pipeline itself (e.g., strong authentication, access privilege reviews, branch protection rules, build rules). These security controls also require continuous monitoring to ensure that they are not disabled. Security teams must arm themselves with constant monitoring and immediate notification should anyone or anything tamper with these settings, which would signal potential malicious activity.

Cycode: Pioneering Solution for Software Development Pipeline Security

Cycode is a complete software supply chain security solution that provides visibility, security, and integrity across all phases of the SDLC. Cycode software integrates with DevOps tools and infrastructure solutions, hardens their security postures by implementing consistent governance, and reduces the risk of breaches with a series of scanning engines that look for issues like hardcoded secrets, infrastructure as code misconfigurations, and code leaks. Cycode's knowledge graph tracks code integrity, user activity, and events across the SDLC to prioritize risk, find anomalies, and prevent code tampering.

In addition to helping customers implement consistent security policies across their SDLC teams and tools, Cycode takes a platform approach that uses multiple complementary security techniques to tackle the myriad of attack vectors that could result in a software supply chain breach, similar to those experienced by Solarwinds and Kaseya. Not only does this approach close entry points that attackers could use, it produces synergies by tackling these problems from one platform that is capable of tracking context across the SDLC phases. The result is better accuracy and greater risk reduction than possible using a collection of similar point products, each of which addresses specific aspects of software supply chain security yet leaves the overall pipeline vulnerable to attack. Cycode is an end-to-end solution that closes the SDLC pipeline gaps and empowers users to implement consistent security.





Growth is a journey. We are your guide.

For over six decades, Frost & Sullivan has provided actionable insights to corporations, governments and investors, resulting in a stream of innovative growth opportunities that allow them to maximize their economic potential, navigate emerging Mega Trends and shape a future based on sustainable growth.

Contact us: [Start the discussion](#)