Adaptive Risk-based Enforcement using SBOM Automation for Secure Software Supply Chains

Sri Sowmya Nemani
Independent Cybersecurity Researcher
San Jose, California

ABSTRACT
Nowadays, many developers rely on third-party and open-source libraries that integrate directly into production software. However, it is critical to understand what is being integrated and who maintains it. The hidden security and governance risks within unmanaged dependencies continue to expose organizations to software supply chain attacks and compliance violations. Software Bills of Materials (SBOMs) in formats such as SPDX and CycloneDX — provide visibility into third-party components. This paper discusses how SBOMs can be automatically generated from development code and integrated into CI/CD pipelines for continuous risk assessment. The model proposed in this study ensures that every building produces an auditable SBOM, allowing the security team to continuously review, mitigate, or apply compensating controls for identified risks.

Keywords
Software Bill of Materials (SBOM), Supply Chain Security, CI/CD, DevSecOps, Risk Mitigation, CMDB, Infrastructure as Code (IaC)

1. INTRODUCTION
The goal of this paper is to integrate an Adaptive SBOM Model within CI/CD pipelines to identify third-party risks early in the development lifecycle and take context-based actions such as applying compensating controls or enforcing automatic mitigation.

This study presents a practical implementation model developed by the author, using open-source tools for SBOM generation and integration with Configuration Management Databases (CMDB) and Infrastructure as Code (IaC) systems. This enables full traceability of software components, version changes, and risk posture across the lifecycle of each building.

2. RESEARCH QUESTIONS
The rapid adoption of open-source and third-party packages has accelerated software innovation but simultaneously increased the risk of supply chain compromise. From SolarWinds to NPM. To address these risks, organizations are increasingly adopting the Software Bill of Materials (SBOM), a comprehensive inventory that lists all components, versions, and licenses used in a software system. However, in most organizations, SBOMs are generated manually or post-release, limiting their usefulness for real-time risk mitigation. To support shift left and identify if the organization is affected, it is important to have the SBOM inventory generated in the non-production environment. Research Question: How can automated SBOM generation and adaptive enforcement within CI/CD pipelines enhance visibility, reduce supply chain risks, and support continuous compliance across software systems?

2.1 EXAMPLES
A very recent incident in the JavaScript ecosystem about the npm supply chain attack started with a phishing attack and escalated to a supply chain attack affecting millions of developers and cloud environments that rely on these libraries. [1][8] Similarly, SolarWinds was hacked, and a total of 18,000 customers and businesses were impacted. The attack was traced back to a malicious software update added to SolarWinds' Orion software, demonstrating the importance of secure software updates in the supply chain.[2][9]

3.METHODOLOGY
The methodology adopted in this research follows a practical implementation model based on the open-source project "Secure OSS Compliance Release Automation Pipeline" [3]. The model integrates automated Software Bill of Materials (SBOM) generation, vulnerability assessment, and license-based policy enforcement within a Continuous Integration/Continuous Deployment (CI/CD) environment. This approach enables adaptive security enforcement and continuous compliance monitoring across every stage of the software build lifecycle. [3][10]

3.1 EXPERIMENTAL ENVIRONMENT AND SETUP
The experimental setup utilized containerized environments orchestrated through GitHub Actions for reproducible CI/CD workflows. Test data consisted of 50 open-source software repositories written in Python, JavaScript, and Go, representing diverse dependency ecosystems. Each repository was automatically built, scanned, and analyzed to generate SBOMs and vulnerability reports. The environment included the following major tools and components:

TOOL	PURPOSE	OUTPUT	VERSION
syft	SBOM generation	SPDX/ CycloneDX	v.0.94.0
Trivy	Vulnerability and License scanning	JSON/ Table	v.0.56.2
Semgrep	Static Analysis (SAST)	JSON	V 8.20
ZAP	Dynamis testing (DAST)	HTML/XML	V 2.14
Conftest	PolicyEnforcement	Rego Rules	V 0.51

3.2 SBOM GENERATION WORKFLOW

The workflow begins once a developer commits code to the repository. The CI/CD pipeline automatically initiates a series of build jobs that generate and analyze SBOM data. Using Syft, an SBOM is created from both the application source code and the resulting container image. The output is exported in CycloneDX JSON format, providing a machine-readable inventory of components, versions, and licenses.

3.3 PIPELINE ARCHITECTURE

The automated pipeline consists of three main phases:

1. Static Analysis Phase – Runs Semgrep and Gitleaks to identify hardcoded secrets and insecure coding patterns.
2. Build and SBOM Phase – Generates SBOMs using Syft, followed by Trivy scans for vulnerabilities and license compliance.
3. Dynamic and Enforcement Phase – Executes OWASP ZAP for dynamic testing and applies OPA policies for adaptive enforcement.

3.4 LIMITATIONS

While the proposed system effectively automates risk detection, certain limitations exist. The analysis currently focuses on open-source components and may not capture closed-source or proprietary dependencies. Additionally, dependency graphs generated by Syft may vary across ecosystems, leading to inconsistent depth of component mapping. Future versions aim to integrate machine-learning-based anomaly detection to dynamically adjust enforcement thresholds and minimize false positives.

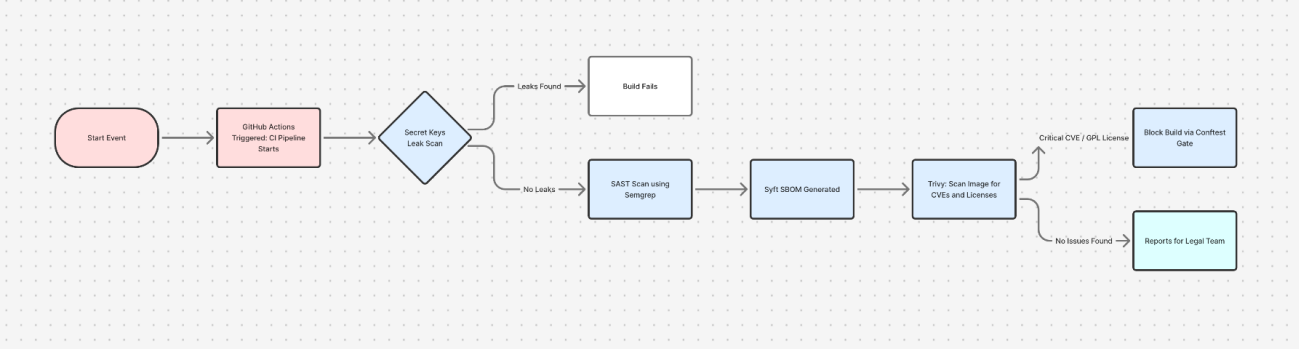

Fig 1: SBOM Workflow

4.CMDB AND IAC

Many organizations still treat SBOMs as passive inventories. This research seeks to close that gap by operationalizing SBOMs into adaptive enforcement pipelines. One of the critical outcomes of the proposed model is linking the SBOM data to enterprise asset management and configuration systems—specifically, the Configuration Management Database (CMDB) and Infrastructure-as-Code (IaC) repositories

5. SOFTWARE AS A SERVICE BILL OF MATERIALS (SaaSBOM)

SaaSBOMs provide an inventory of services, endpoints, and data flows and classifications that power cloud-native applications. SBOMs primarily describe the open-source and third-party software components integrated within an application's source code or build artifacts. However, SaaSBOM provides an inventory of all cloud and service dependencies that power an application. As it includes Services and microservices, Endpoint URL, etc. [7]

SaaSBOMs complement SBOMs and IaC by bridging application-level transparency and service-level dependency management, enabling comprehensive risk governance for DevSecOps pipelines.

6. DISCUSSIONS

The results of this study demonstrate that automating SBOM generation and enforcement within CI/CD pipelines substantially improves software supply chain visibility and reduces dependency risk. Unlike static vulnerability assessments performed post-release, the proposed adaptive model allows organizations to detect and mitigate risks early in the software development lifecycle (SDLC)

By integrating open-source tools such as Syft and Trivy, this approach aligns with modern DevSecOps principles, enabling a "shift-left" security posture that treats compliance and vulnerability checks as continuous and automated.

However, Strict enforcement may introduce business interruptions when the pipeline blocks build due to critical vulnerabilities or license conflicts, and Adaptive enforcement policies, where the build system differentiates between critical, medium, and low-risk issues, help mitigate this tension.

7. CONCLUSIONS

This paper presented an Adaptive SBOM Enforcement Model designed to integrate with CI/CD pipelines for real-time security and compliance validation. Using open-source tools and automation, the model generates both SBOMs (software-level visibility) and SaaSBOMs (service-level visibility), performs vulnerability and license checks, and applies policy-driven enforcement.

8. LITERATURE REVIEW

Search Strategy: The literature search included peer-reviewed articles, industry white papers, and government advisories focused on SBOMs, software supply-chain security, and CI/CD automation. Databases searched included SpringerLink, ResearchGate, OWASP guides, and CISA advisories (2020–2025).Towards a More Secure Ecosystem: Implications for Cybersecurity Labels and SBOMs: This paper targets two related efforts to create more transparency in the global software supply chain: labels and Software Bills of Materials.[5] The Impact of SBOM Generators on Vulnerability Assessment in Python: A Comparison and a Novel Approach : This paper talks about the SBOM inventory

vulnerability assessment, the first security analysis on the vulnerability detection capabilities of tools receiving SBOMs as input. We comprehensively evaluate SBOM generation tools by providing their outputs to vulnerability identification software. [6]

Future work will focus on scaling this model across multi-cloud and enterprise-grade CI/CD environments while integrating AI-driven anomaly detection and predictive analytics for risk scoring. Additional studies using diverse datasets and programming ecosystems will further validate the generalizability and performance of the adaptive SBOM framework.

9. REFERENCES

[1] Palo Alto Networks, "NPM Supply-Chain Attack," Cloud Security Blog, 2025.

[2] SolarWinds, "An Investigative Update of the Cyberattack," Technical Report, 2025.

[3] Nemani, S., "Secure OSS Compliance Release Automation Pipeline," GitHub Repository, 2025.

[4] Anchore, "How Syft Scans Software to Generate SBOMs," Technical White Paper, 2024.

[5] Camp, L., "Towards a More Secure Ecosystem: Implications for Cybersecurity Labels and SBOMs," ResearchGate, 2023.

[6] Springer, S., *"The Impact of SBOM Generators on Vulnerability Assessment in Python," Springer LNCS*, 2024.

[7] OWASP Foundation, *"CycloneDX Authoritative Guide to SBOM,"* 2024.

[8] Cybersecurity and Infrastructure Security Agency (CISA), *"Widespread Supply Chain Compromise Impacting NPM Ecosystem,"* Alert Bulletin, 2025.

[9] Center for Internet Security (CIS), *"SolarWinds Incident Overview,"* 2025.

[10] Secure by Design, *"CI/CD Hardening Guide,"* Implementation Handbook, 2024.