RUNSAFE
SECURITY

Reducing Risk in Your Software Supply Chain:

Addressing Open Source, Emerging Threats, and Regulatory Shifts

RunSafeSecurity.com

Table of Contents

Introduction

Software supply chain attacks have risen by over 700% in the past few years. With incidents like SolarWinds, Log4j, and Okta making headlines, the ripple effects of a successful breach have become widely known.

The software supply chain is vulnerable for several reasons. Software today passes through many hands in its journey from design and development to deployment and maintenance. Open-source communities, commercial vendors, in-house developers, and now AI-generated code all contribute to a final product delivered to the market.

While this ecosystem has many benefits, the complexities make it challenging to have a full picture of the software and any vulnerabilities that lie within. We've seen cybercriminals and nation-state actors specifically target the software supply chain to exploit vulnerabilities for espionage, disruption, or financial gain. Every breach reveals the deep interdependencies and fragility of the current ecosystem, with one successful attack often affecting companies and devices across industries.

The risks are there. What steps should organizations take to safeguard their software?

In this white paper, we will cover:

- The threat landscape, recent high-profile attacks, and regulatory and industry developments in improving software supply chain security
- Mitigation strategies to reduce software supply chain risk
- A case study of how RunSafe Security helped a digital infrastructure provider secure its software supply chain

Top Software Supply Chain Risks

At the core of the software supply chain lies the intricate dance of code creation and dissemination. Developers leverage tools, libraries, and frameworks to craft software solutions tailored to meet the demands of their users.

At each stage of the software supply chain lifecycle, adversaries can exploit potential points of vulnerability to infiltrate systems, compromise data, or disrupt operations. These vulnerabilities may arise from insecure coding practices, outdated software components, or malicious actors inserting backdoors or malware into software packages. The biggest software supply chain risks include the following.

1. Third-Party Vulnerabilities

Third-party software components play a pivotal role in accelerating development cycles and enhancing software functionality. But these dependencies also serve as potential entry points for malicious actors seeking to exploit vulnerabilities. Organizations face the daunting challenge of securing an ever-expanding attack surface without mechanisms for vetting and monitoring third-party components.

A REPORT BY DATA THEOREM FOUND THAT

91%

of organizations experienced a **software supply chain attack in 2023**

Of those security incidents

41%

involved a **zero-day exploit** on vulnerabilities within third-party code.

2. Open-Source Software Risks

Open source code is integral to modern software. According to a 2023 survey by the Linux Foundation, 90% of organizations globally use open source software (OSS) to a moderate, significant, or widespread extent. While OSS has many benefits and in many cases leads to more secure code bases, when a breach does occur, the effects are particularly severe and far-reaching.

Threat actors exploit widely used libraries to maximize their reach, as seen in the 2024 XZ Utils backdoor incident, where a supply chain attack inserted a sophisticated backdoor into a widely used Linux compression library. This backdoor could enable remote code execution on affected systems.

The decentralized nature of OSS communities poses inherent risks in terms of vulnerability management and code integrity. As organizations integrate OSS into their projects, they must navigate the complexities of patch management, license compliance, and code hygiene to mitigate the risk of exploitation.

3. Build Environment Security

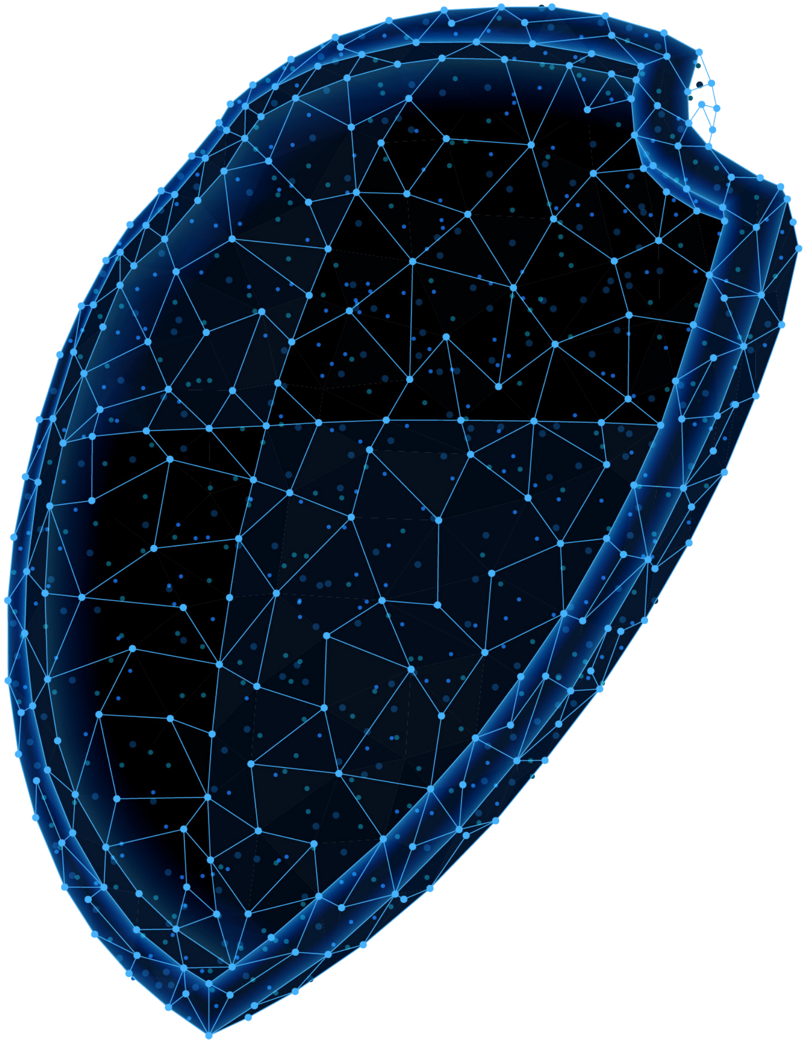
Securing the build environment is critical. Compromising build servers or CI/CD pipelines can enable attackers to inject malicious artifacts without directly modifying the source code. The 2023 CircleCI breach, caused by compromised developer credentials, highlighted how attackers can move laterally across build environments and exfiltrate secrets, tokens, and private keys.

4. AI-Driven Risks in the Software Supply Chain

The integration of artificial intelligence (AI) into software development and deployment pipelines has introduced a new class of supply chain risks. AI now plays a role not only as a tool for automation and code generation but also as a potential attack vector and source of vulnerabilities.

AI-generated code may contain exploitable vulnerabilities or reference non-existent or malicious third-party libraries, creating new opportunities for supply chain attacks. If such code is shared in public repositories and reused, vulnerabilities can propagate rapidly across the ecosystem.

Additionally, the proliferation of open-source AI models and tools has made it easier for attackers to distribute malicious components disguised as legitimate machine learning utilities. For example, security researchers have identified malicious packages uploaded to popular repositories like PyPI, which, when included in AI projects, can exfiltrate data or provide remote access to attackers



The Current Landscape

The convergence of nation-state tactics with financially motivated cybercrime means attackers are well-resourced, patient, and highly adaptive. Supply chain attacks are shifting from opportunistic exploits to strategic infiltration over time.

Trends to watch:

- Insider and “sleeper” contributor threats (as seen in XZ Utils)
- Exploitation of build systems and containers
- Exploitation of widespread memory corruption vulnerabilities in unmanaged codebases
- Dissemination of malicious packages and components in open-source code and AI-generated code and AI models

High-Profile Software Supply Chain Attacks and Their Impact

Two notable attacks that reverberated throughout the cybersecurity community are the SolarWinds attack and Log4j vulnerability. More recent attacks include the CircleCI breach, MOVEit transfer, and XZ Utils Backdoor. The incidents highlight the need for heightened vigilance in identifying and patching vulnerabilities in software components, as well as implementing strategies to mitigate the risk of exploitation.

SolarWinds (2020)

The SolarWinds attack, discovered in late 2020, involved threat actors compromising the SolarWinds Orion platform, a widely-used network management system. By injecting malicious code into the supply chain of SolarWinds, attackers gained unauthorized access to thousands of organizations, including government agencies and major corporations. This attack had far-reaching consequences, leading to data breaches, espionage activities, and significant financial losses.

On average, companies impacted by the SolarWinds attack reported losses equal to 11% of their annual revenue.

Log4j (2021)

Similarly, the Log4j vulnerability, discovered in December 2021, exposed critical vulnerabilities in the Apache Log4j library, a widely-used logging tool in Java-based applications. Exploiting this vulnerability threat actors could execute arbitrary code remotely, potentially compromising sensitive data and systems.

The widespread use of Log4j across various software applications amplified the impact of this vulnerability, affecting organizations worldwide and highlighting the challenges of securing the software supply chain.

CircleCI Breach (2023)
Compromised developer tokens led to a breach of CI/CD infrastructure, affecting customers' secrets and credentials. The incident stressed the importance of secure build environments and secrets management.

MOVEit Transfer Exploit (2023)
The ransomware group Cl0p exploited vulnerabilities in MOVEit Transfer, a widely used secure file transfer tool. The attack affected over 620 organizations, including major airlines, media companies, and government organizations, compromising large volumes of personal and sensitive data. The attack reinforced the need for aggressive vulnerability management in widely used enterprise tools.

Okta Supply Chain Attack (2023)
Threat actors accessed Okta's customer support management system using compromised credentials, stealing private customer data and files uploaded in support cases. The breach went undetected for weeks, impacting Okta's extensive customer base.

JetBrains TeamCity Attack (2023)
Attackers exploited a critical authentication bypass vulnerability in JetBrains TeamCity CI/CD servers, allowing remote code execution and administrative control. The flaw was reportedly exploited by Russian threat actors (Cozy Bear), with over 3,000 vulnerable servers exposed online.

XZ Utils Backdoor (2024)
A contributor inserted malicious code into a compression library used by major Linux distributions. The backdoor enabled unauthorized SSH access to systems running compromised versions. This attack revealed how low-level utilities can become vectors for mass compromise.

Malicious npm Packages (2024)
Attackers uploaded malicious npm modules (e.g., warbeast2000, kodiak2k) to GitHub, which stole SSH keys from infected developers and, in some cases, attempted to extract passwords using Mimikatz. Hundreds of developers were affected before the packages were removed.

Regulatory Developments and Industry Standards

In response to the growing threat landscape, regulatory bodies and industry organizations are issuing recommendations and compliance requirements to strengthen software supply chain security.

Regulatory frameworks are adding requirements for a Software Bill of Materials (SBOM) to enhance transparency and accountability in the software supply chain while industry standards emphasize the importance of risk management, threat intelligence sharing, and secure development practices in mitigating supply chain risks.

Executive Order 14028 (U.S.): Mandates SBOMs for federal software suppliers and encourages adoption of secure development practices.

NIST Secure Software Development Framework (SSDF): Guides organizations in implementing secure design, development, and testing.

EU Cyber Resilience Act (CRA): Places strict cybersecurity obligations on hardware and software vendors, including requirements for SBOM generation.

Industry-Led Initiatives: The Open Source Security Foundation (OpenSSF), for example, promotes security hygiene in open-source projects.

Mitigation Strategies for Strengthening Software Supply Chain Security

As threats to the software supply chain continue to escalate, organizations must adopt comprehensive strategies to mitigate risk and enhance resilience. Key focus areas include securing the development lifecycle, managing third-party components, improving risk assessments, and increasing transparency through Software Bills of Materials (SBOMs).

Securing the Development Lifecycle

Integrating security into every stage of the Software Development Lifecycle (SDLC) is key to reducing vulnerabilities and strengthening software from the ground up. Security best practices should be incorporated from design through deployment, supported by:

- Developer training in secure coding techniques
- Automated tools for code analysis and vulnerability scanning
- Early identification and remediation of risks before they reach production

Adopting methodologies like DevSecOps ensures that security is embedded throughout the development process—not just applied at the end. This approach fosters a culture of shared responsibility for security and reduces the likelihood of flaws slipping through.

RunSafe supports DevSecOps by integrating with Continuous Integration (CI) tools, enabling security protections to be automatically applied during the build process. Whether using automated systems like Yocto or Buildroot or conducting manual builds, RunSafe seamlessly embeds runtime protections into the software.

For example, in Yocto-based builds—which pull both proprietary and open-source code to create fresh builds—RunSafe adds a security layer directly into the process. This ensures that up-to-date protections are built into every component.

Additionally, RunSafe supports a wide range of compilers and operating systems, including embedded Linux, Android, VxWorks, and LynxOS. This broad compatibility allows organizations to secure applications across diverse environments with minimal changes to existing workflows.

Improving Third-Party Vendor Management

Third-party components are a cornerstone of modern software development, but they also introduce risk. Effective vendor management is critical to minimizing these risks.

Organizations should apply rigorous security standards when selecting and evaluating third-party vendors, including:

- Security track record
- Compliance with industry regulations
- Responsiveness to vulnerability disclosures

Implementing Continuous Risk Assessments

The global cost of cybercrime is projected to reach
$10 TRILLION ANNUALLY BY 2025,
with supply chain vulnerabilities being a major contributor

Continuous risk assessments are crucial for identifying and mitigating vulnerabilities throughout the software supply chain. Organizations must evaluate risks across all software sources, including internally developed code, third-party components, and open-source libraries.

Understanding who maintains a software component and how frequently it's updated is critical. Actively maintained projects with many contributors and frequent updates tend to be more secure and reliable. In contrast, less active or poorly maintained projects often pose higher risks due to limited resources for patching vulnerabilities.

To manage this complexity, organizations should establish a dynamic risk assessment framework that considers:

- The criticality of each component
- Its potential impact
- The likelihood of exploitation

This framework should evolve in response to the changing threat landscape, ensuring that organizations remain aware of emerging risks and can prioritize resources accordingly. A proactive approach allows for early identification of vulnerabilities and more effective mitigation strategies.

By broadening the scope of risk assessment to all software sources, incorporating contributor and update activity, and using clear, adaptable evaluation criteria, organizations can better manage risk and strengthen their software supply chains.

Enhancing Transparency with SBOMs

SBOMs (Software Bills of Materials) are a foundational element in software supply chain security. They provide an inventory of software components and dependencies to give organizations visibility into potential vulnerabilities and risks in software. SBOMs are quickly becoming industry best practice and required by regulators in many circumstances, like with the FDA in the United States and Cyber Resilience Act in the EU.

RunSafe stands apart by generating SBOMs during the build process, not after the fact. Unlike traditional approaches that analyze final binaries using heuristics—which can miss critical dependencies—RunSafe's method captures every component, library, and process involved in software creation.

This build-time approach ensures a more accurate and comprehensive SBOM, revealing second-order dependencies and other elements that may otherwise go unnoticed. This level of detail allows organizations to:

- Rapidly identify and respond to vulnerabilities
- Maintain stronger control over their software composition
- Communicate transparently with stakeholders and regulators

By improving visibility and precision in SBOM generation, RunSafe empowers organizations to proactively manage risk and demonstrate compliance with industry and regulatory standards.

PREPARING FOR AI

The integration of artificial intelligence into software development workflows has introduced novel attack vectors that require specialized security approaches. AI-generated code may contain exploitable vulnerabilities or reference non-existent libraries, creating opportunities for supply chain attacks that can rapidly propagate across the ecosystem. Organizations must implement dedicated code review processes specifically designed to identify AI-generated vulnerabilities, including automated verification systems that check all dependencies suggested by AI tools against trusted repositories and scan for hallucinated dependencies or suspicious code patterns.

Additionally, organizations should establish comprehensive governance frameworks for AI use in software development, including rigorous vetting processes for AI coding assistants and development tools. This includes verifying the provenance of AI models, assessing the security posture of AI service providers, and maintaining inventories of all AI tools used in development processes. Teams should implement approval workflows for AI-generated code integration and establish clear boundaries for AI tool usage in critical system components, ensuring that human oversight remains central to security-critical development decisions.

IMPLEMENT BUILD ENVIRONMENT HARDENING

The compromise of build infrastructure represents one of the most effective attack vectors for supply chain infiltration, as demonstrated by incidents like the CircleCI breach and JetBrains TeamCity attack. Organizations must implement comprehensive security measures including network segmentation of build systems, immutable build environments that are created fresh for each build process, and robust secrets management with just-in-time access principles. Build environments should be treated as high-value targets with dedicated security controls, intrusion detection systems, and continuous monitoring for unusual activity such as unexpected network connections or anomalous resource usage.

Critical to build security is implementing build attestation and provenance tracking to ensure the integrity of software artifacts throughout the development pipeline. Organizations should deploy ephemeral build agents where possible, maintain strict access controls to build infrastructure, and implement automated rotation of secrets and credentials used in build processes. Additionally, comprehensive logging and behavioral analysis tools should be deployed to detect deviations from normal build patterns and alert security teams to potential compromises before malicious artifacts can be distributed to production environments.

Case Study: Critical Digital Infrastructure Provider

As mentioned, organizations face myriad challenges in securing their supply chains against emerging threats. Some forward-thinking companies have successfully navigated these complexities by adopting RunSafe Security solutions.

A critical digital infrastructure provider encountered a common dilemma many organizations face: disparate development teams using varied tools and methodologies, resulting in inconsistent security postures across products.

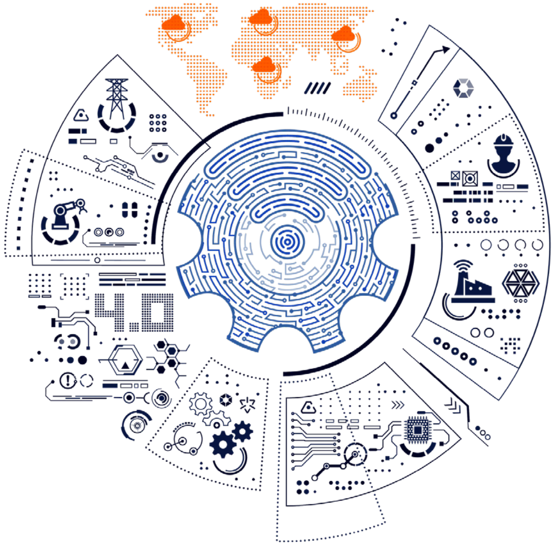
Recognizing the need for standardized approaches to enhance security, the security team initiated a comprehensive overhaul of its development processes, focusing on standardizing toolsets and streamlining workflows to improve efficiency. By selecting a single CI tool and implementing centralized vulnerability identification within its CI environment, the company achieved greater visibility and control over its software supply chain.

The key to success was the adoption of RunSafe Security's solutions, which included generating a build-time Software Bill of Materials (SBOM) and vulnerability scoring. Integrating these capabilities into their CI toolchain empowered developers to make informed decisions about mitigating vulnerabilities, whether through manual code fixes or automated protections.

The result?

Implementing RunSafe strengthened their product's security posture and optimized developer productivity. This gives developers the tools and insights to prioritize security tasks efficiently, enabling their team to focus on innovation and customer satisfaction rather than being bogged down by endless vulnerability remediation efforts.

This strategic partnership with RunSafe Security exemplifies the transformative impact of proactive security measures in the software supply chain. The company fortified its defenses against evolving threats by standardizing processes and centralizing vulnerability management while driving business agility and innovation.



Building Resilient Software

The software supply chain has become the critical battleground where cybersecurity is won or lost. As this white paper has demonstrated, the 700% increase in supply chain attacks reflects not just growing threat actor sophistication, but the fundamental reality that our digital infrastructure has become inextricably interconnected. From the SolarWinds breach that compromised thousands of organizations to the XZ Utils backdoor that nearly infiltrated major Linux distributions, we've witnessed how a single point of compromise can cascade across entire ecosystems.

The convergence of multiple risk factors—third-party dependencies, open-source proliferation, AI-generated code, and vulnerable build environments—has created a perfect storm of supply chain vulnerability. Traditional perimeter-based security approaches are inadequate when the threat originates from within the very components we trust. The challenge is no longer simply identifying vulnerabilities after they occur, but preventing exploitation of the fundamental weaknesses that make supply chain attacks possible in the first place.

RUNSAFE
SECURITY

ABOUT RUNSAFE SECURITY, INC.

RunSafe Security protects embedded software across critical infrastructure, delivering automated vulnerability identification and software hardening from build-time to runtime to defend the software supply chain and critical systems without compromising performance or requiring code rewrites. The RunSafe Security Platform includes the authoritative build-time SBOM generator for embedded systems and C/C++ projects, automated vulnerability identification and risk quantification, patented memory relocation techniques to mitigate memory-based vulnerabilities, and pre-hardened open-source packages and containers for immediate protection.

Headquartered in McLean, Virginia, with an office in Huntsville, Alabama, RunSafe Security's customers span the aerospace and defense, energy, operational technology, industrial automation, transportation and automotive, medical device, and high-tech manufacturing verticals.

RunSafeSecurity.com
☎ 571.441.5076
Sales@RunSafeSecurity.com

Strengthening Software Supply Chain Security with RunSafe

The software supply chain security challenges outlined in this white paper—from third-party vulnerabilities and open-source risks to AI-driven threats and build environment compromises—require comprehensive, proactive solutions that address vulnerabilities at their source. RunSafe Security provides a transformative approach that directly tackles these interconnected risks while seamlessly integrating into existing development workflows.

Complete Supply Chain Visibility

RunSafe's build-time SBOM generation delivers unprecedented supply chain visibility by capturing every component, library, and dependency during compilation—revealing direct and transitive relationships that traditional binary analysis tools often miss. This comprehensive approach enables organizations to respond in near real-time to emerging threats like Log4j-type vulnerabilities, while proactive vulnerability scoring and exposure quantification transform reactive management into strategic risk mitigation. The platform's runtime protection technology automatically hardens software against memory corruption exploits during the build process, effectively neutralizing the majority of critical vulnerabilities in C and C++ codebases without requiring source code modifications.

Seamless DevSecOps Integration

RunSafe's broad compatibility with compilers, operating systems, and build environments—including embedded Linux, Android, VxWorks, LynxOS, Yocto, and Buildroot—ensures that organizations can implement consistent security controls across diverse technology stacks. Automated integration with CI/CD pipelines eliminates human error and process gaps, while developer-friendly implementation typically requires less than a day to deploy. This seamless DevSecOps integration means security protections are automatically applied to every build without disrupting established workflows.

Measurable Business Impact

RunSafe delivers measurable business impact through reduced vulnerability burden, accelerated innovation cycles, regulatory compliance support, and clear risk quantification metrics. By embedding security directly into the build process and providing comprehensive supply chain visibility, organizations can strengthen their security posture while enabling developers to focus on feature development rather than endless vulnerability patching—creating a foundation of protection that travels with software throughout its entire lifecycle.