Scribe

FROM APPLICATION SECURITY TO SOFTWARE SUPPLY CHAIN SECURITY:

A Fresh Approach Is Needed

TABLE OF CONTENTS

INTRODUCTION

The traditional approach to securing software products focuses on eliminating vulnerabilities in custom code and safeguarding applications against known risks in third-party dependencies. However, this method is inadequate and fails to address the full scope of threats posed by the software supply chain.

Neglecting to secure every aspect of this chain, from production to distribution and deployment, exposes organizations to attacks such as malware, data breaches, and intellectual property theft. Ignoring this critical step is a serious disregard for the organization's security.

In this white paper, we will examine the rising trend of cyber attacks directed at the software supply chain, as well as recent regulatory advancements and best practice frameworks that have arisen in response to this growing danger. We will also shed light on the need for a fresh strategy to secure your software supply chain, one that surpasses current application security measures.

We will illustrate why current investments in application security offer some protection but do not completely mitigate your cyber security risks in this area. Finally, we will outline what is necessary to complement them for full protection.

WHY SECURING THE SOFTWARE SUPPLY CHAIN IS IMPORTANT

The threat to software supply chains

The use of third-party tools, libraries, and open-source software in software development increases both the complexity of the software supply chain and the risk of potential vulnerabilities and attacks. Attackers can target a specific link in the supply chain to gain access to sensitive information or disrupt operations.

These attacks can have a significant impact on organizations and their supply chain partners.

Some well-known security incidents or attacks that have been reported in recent years:

Passwordstate
Enterprise Password Management

Passwordstate:

A password management tool that suffered a data breach in 2020, involving attackers compromising the tool's update mechanism and planting a malicious update that exposed sensitive information of its users.

solarwinds

SolarWinds:

A massive cyber attack that was discovered in December 2020, affecting multiple organizations and government agencies. The attackers utilized a supply chain attack to infiltrate the software of SolarWinds, a popular IT management software.

Codecov

Codecov:

A popular code coverage analysis tool suffered a supply chain attack in 2021, where attackers were able to access sensitive information from Codecov's customers.

The fallout from several software supply chain attacks has made headline news in recent years. As a consequence, Gartner has predicted recently that by 2025, 45% of organizations worldwide will have experienced attacks on their software supply chains, a three-fold increase from 2021.

As software supply chain security gains attention, various application security solution vendors are rebranding themselves as offering solutions in this space. But is software supply chain security simply a new term for traditional application security, or is it a distinct market segment with unique characteristics and technologies? Is having an Application Security (AppSec) program in place sufficient to address software supply chain security, or do organizations need to adopt separate standards and technologies to ensure the security of their software and its supply chain? This question will be explored further in the white paper.

Emerging regulation and best practices frameworks

The U.S. Executive Order on Cybersecurity and other similar regulations reflect the growing importance of software supply chain security and the need for organizations to take appropriate measures to protect their software and its supply chain. The increasing interest shown by major players in collectively introducing measures to combat the threat is a positive development and indicates a growing recognition of the need for a coordinated and industry-wide approach to address the growing threat.

Some of the key initiatives to address software supply chain security include:

Enhancing software development practices:

Incorporating security best practices during the software development life cycle and conducting regular security assessments.

Improving supply chain visibility:

Having a clear understanding of the components and suppliers involved in the software supply chain and ensuring that they meet security standards.

Implementing secure software procurement practices:

Conducting due diligence on software vendors and suppliers and incorporating security requirements into procurement processes.

Developing threat intelligence capabilities:

Collecting, analyzing, and sharing information about potential threats to the software supply chain.

Overall, these initiatives aim to increase the security of software and the supply chain by reducing the risk of attacks and ensuring that software is free from vulnerabilities and malicious code.

Below is a brief overview of the key initiatives:

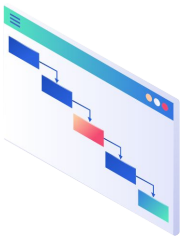
1. NIST's Secure Software Development Framework (SSDF)

NIST SP 800-218 represents a key development in ensuring software supply chain security for organizations supplying software and software services to the U.S. government. It provides a set of secure development practices that can be integrated into organizations' software development life cycle (SDLC) to reduce the number of vulnerabilities in released software and prevent future occurrences. The guidelines are outcome-focused, customizable, and sector-agnostic, making them suitable for organizations involved in both software development and acquisition.

For a deeper dive into the SSDF and the impact it will have on the software industry, check out this whitepaper.

2. The United States Office of Management and Budget (OMB) Memo

In 2022, the OMB released two memos that further emphasize the importance of software supply chain security and the role of Software Bills of Materials (SBOMs) in ensuring the security and integrity of the software supply chain. With the binding timeline for implementation by 2024, organizations must take action to comply with the requirements outlined in the memos and ensure the security of their software and its supply chain. Additionally, federal agencies must take action to implement guidelines, including sharing information with the private sector and obtaining artifacts from vendors demonstrating secure software development practices.



3. Supply Chain Levels for Software Artifacts (SLSA) Framework

The SLSA framework is a comprehensive set of security controls and standards designed to ensure the integrity of software supply chains. It was developed by OpenSSF, Google, and other cybersecurity stakeholders. By following this end-to-end framework, unauthorized changes to software packages can be prevented. Adopting SLSA can help protect against common supply chain attacks.

4. The EU Cyber Resilience Act

The EU Cyber Resilience Act aims to improve the cyber resilience of organizations in the EU. It requires organizations to make SBOMs available for all software products they use in order to help them identify and address potential security vulnerabilities early on. The act also requires organizations to report vulnerabilities to ENISA/CERT-EU. It provides a framework for improving the security of software products used and for managing the associated risks. By requiring SBOMs, the act will improve the overall security posture of organizations in the EU and increase their ability to respond to and recover from cyber-attacks.

5. CIS

The Center for Internet Security (CIS) developed the CIS Software Supply Chain Security Guide, which covers the phases of the software supply chain, from code contribution to delivery to end consumers. The guide provides best practices for securing the software supply chain, including establishing a security program, implementing secure development practices, conducting regular security assessments of suppliers, implementing security controls for distribution and deployment, and continuously monitoring for vulnerabilities and attacks.

6. Sigstore

Sigstore is an open-source project focused on securing software supply chains. It provides a method for better-securing software supply chains in an open, transparent, and accessible way. The key to securing software supply chains is digitally signing the various artifacts of which applications are comprised. Sigstore aims to make software signing ubiquitous and easier by providing a simplified and quicker solution, compared to traditional digital signing solutions. The project also creates an open and immutable activity log.

THE ROLES OF THE DIFFERENT STAKEHOLDERS

Software supply chain security is a multi-faceted effort that involves the collaboration of several parties including AppSec, DevOps, and DevSecOps, compliance officers, legal professionals, and production security. AppSec focuses on identifying and mitigating security risks in the development and deployment of software applications. Compliance officers define policies and procedures to protect software and systems, DevOps/DevSecOps enforce them, and legal professionals ensure license compliance with legal and regulatory requirements. Production security protects the integrity and availability of the production environment and ensures that all software and systems are up-to-date with the latest security patches and updates.

The goal and tasks of the AppSec discipline

The AppSec team in mature software producer organizations focuses on securing software through developer training, code review, automatic scanning, and monitoring dependencies. The most common methodologies are static and dynamic application software testing (SAST and DAST) for testing applications, and software composition analysis (SCA) to detect known vulnerabilities in open-source dependencies.

To cope with the evolving threat landscape, several newer solutions have been devised and adopted. One such solution is the use of secret detection tools, which aid organizations in identifying secrets such as credentials, API keys, and sensitive data that may have been unintentionally disclosed in the code. Additionally, new scanning tools have been created to detect vulnerabilities in containers and infrastructure-as-code (IaC) in cloud computing environments, allowing organizations to ensure the security of their applications as they transition to the cloud.

To manage the vast array of security tools, orchestration solutions have arisen to provide a unified view of security across the SDLC. These solutions centralize the management of security tools and policies, streamlining the security process and reducing the risk of missed vulnerabilities.

Triaging, deduplication, and prioritization of security alerts are among the major challenges faced by AppSec owners. It can be overwhelming to manage the volume of alerts and to determine which ones require immediate attention and which can be disregarded. Additionally, AppSec owners need to assess the impact of the vulnerabilities on their specific systems, which requires a deep understanding of the technology stack and applications.

The AppSec approach covers the SDLC from code writing through build, packaging, and testing and is crucial in reducing vulnerabilities in both custom code and open-source components, thus mitigating liability for software producers and operators. Leading vendors in this space include Veracode, Checkmarx, Snyk, Synopsis, and Fossa.

The diagram below offers a simplified overview of the AppSec market, showcasing several key players. Note that Aqua's presence is due to its acquisition of Argon.

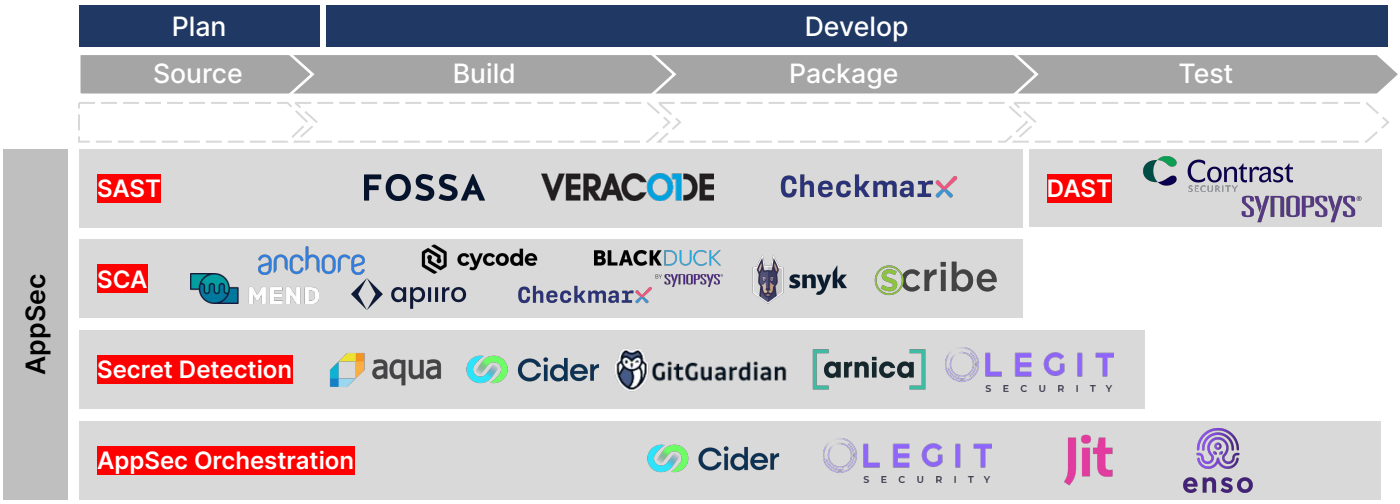


Diagram: A simplified view of the AppSec market landscape

While AppSec solutions can be effective in detecting known vulnerabilities and insecure code patterns, they tend to be reactive rather than proactive; they do not proactively reduce the attack surface.

Moreover, the high volume of false-positive alerts generated by scanners can make it challenging for developers to focus on the most critical security issues. The scanning process can also be time-consuming and disruptive to the development timeline, adding friction to the development process and slowing down the release of new software.

THE EMERGING ROLE OF SOFTWARE SUPPLY CHAIN SECURITY

AppSec solutions overlook areas of the SDLC that may be vulnerable to attack and that affect the software supply chain. These areas include development tools, code repositories, source code management systems, build servers, and artifact registry systems, which may contain vulnerabilities and misconfigurations.

On the other hand, from the software consumers' perspective, a common concern is that third-party software can introduce security vulnerabilities that are difficult to detect and mitigate. It's important for consumers to thoroughly assess and vet the software they use, as well as regularly monitor and update it to reduce these risks.

To overcome these challenges, both software producers and consumers need to adopt a wider approach. Software producers should integrate into all stages of the SDLC and leverage automated tools to address security threats.

- ## CI/CD posture management

  In order to prevent supply chain attacks, it is important for organizations to secure their SDLC. CI/CD posture management helps automate the discovery process and enforce security best practices. Enterprises today struggle to gain visibility into their SDLC and ensure that their infrastructure usage in dev environments is secure.

  Some of the additional key requirements for CI/CD posture management include build server authentication, restrictions on public repositories and writable S3 buckets, and the expiration of security keys.

This helps to prevent unauthorized access to sensitive information and to ensure that security keys are rotated frequently to prevent theft.

It is also important to limit risky development practices that can lead to vulnerabilities in the pipeline. This includes executing third-party resources before verification and referencing images in a build that may be altered externally. By implementing these security measures, organizations can improve the security of their software development processes and reduce the risk of supply chain attacks.

- ## Attestation and Provenance

  Maintaining the security and integrity of software supply chains is done by attestation (verifying the authenticity and integrity of software components) and tracking the provenance (origin, ownership, and custody of components). Through attestation and provenance, organizations can make informed decisions about the trustworthiness of the software they use. This helps to reduce the risk of tampering, vulnerabilities, and inadequate security in the development process while ensuring compliance with industry standards.

  To effectively implement attestation and provenance, organizations must gather evidence from various sources involved in the software development process, including source code managers, CI tools, build servers, container registries, and cluster admission controllers.

The evidence may include the identity of the developer committing the code, proof of code review, file hashes, artifact hashes, the security posture of the source control manager and CI tool, and results of application security scans. Evidence should be cryptographically signed for added security, and stakeholders can apply policies over the attestations to ensure secure development and build processes, validate that no tampering has taken place, and assess compliance with standards such as the SSDF and the SLSA framework.

It is important to note that the collection of evidence, signing of attestations, and creation of provenance traces cannot be done through existing APIs or observability solutions. New agents must be developed to trace the specific type and amount of data generated during the SDLC. These agents must also sign the evidence at the appropriate stage in the process to ensure its integrity and authenticity.

To the same end, beyond the SDLC, both software producers and consumers need to seek means to achieving the following:

- ## Continuous compliance

Continuous verification helps maintain compliance and protect against vulnerabilities even after deployment. This involves continuous monitoring and verifying the security of software components throughout the entire software development and deployment process.

One important tool for continuous verification is a Software Bill of Materials (SBOM). SBOMs provide a comprehensive list of all the software components and their versions used in a software project.

By tracking the software components and their versions, SBOMs facilitate vulnerability scans to identify and address any known vulnerabilities in the software components. This allows organizations to stay ahead of potential security threats and maintain the security of their software systems.

Additionally, SBOMs help organizations comply with various regulations and standards such as the SSDF by providing a clear understanding of the software components used in the software and their licenses.

- ## Share SBOMs and attestations and monitor risk continuously

Making SBOMs available and continuously monitoring potential risks is now a necessary step for maintaining the security and compliance of software systems. By sharing SBOMs, software producers provide consumers with a way to manage related risks and comply with regulations. SBOMs enable scans to discover existing vulnerabilities in components and ongoing monitoring to detect new ones. It can also serve to reject applications with components that lack good reputation scores. The software supply chain consists of various links that can pose a threat. Hence, it is crucial to have a system that not only detects vulnerabilities from direct and indirect dependencies but also offers a security attestation from throughout its entire supply chain.

The above four points are the four cornerstones of Software Supply Chain Security and the criteria for choosing a solution in this field.

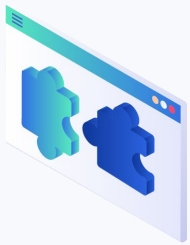
HOW SCRIBE SECURITY ADDRESSES THE CHALLENGE

Scribe is a SaaS solution that helps both software producers and software consumers minimize the risk in their software supply chain.

CI/CD posture management

Scribe connects to the different tools in the software producer's CI/CD pipelines and gathers information it uses to continuously assess their compliance with standards such as the SSDF, CIS, and SLSA, per pipeline, build, or project update.

Attestation and Provenance



Scribe integrates with various tools used in the SDLC using both agent-based and agentless methods. It continuously gathers evidence about the source code, software artifacts, code provenance, and process events, creating an attestation through cryptographic signing. This provides provenance for the built artifacts using Sigstore, Public Key Infrastructure (PKI), and GitHub's GPG keys for signing and verification.

The Scribe agents provide a unique capability to collect evidence, sign attestations, and create provenance traces, which cannot be done through existing APIs or observability solutions. These agents also ensure the integrity and authenticity of the evidence by signing it at the appropriate stage in the process.

Continuous compliance

Scribe produces SBOMs for every build and software product and continuously monitors for vulnerabilities. It alerts on both old and new vulnerabilities from the end of the build to after deployment to production. Scribe also collects information such as reputation and exploitability of dependencies to support risk-based decision-making. The attestations generated from various stages of the build are evaluated for integrity by comparing file hashes, verifying signatures, and checking adherence to secure development and secure build standards.

Share SBOMs and attestations as well as monitor risk continuously

Scribe operates as a secure, cloud-based hub for sharing SBOMs, security information, and attestations among stakeholders. Producers can share this information granularly and in a controlled manner with consumers. Both parties can use Scribe to set and enforce policies and acceptance criteria for SBOMs and their security attestations. Producers can also utilize Scribe to write and share advisories for vulnerabilities using the Vulnerability Exploitability eXchange (VEX) standard for machine-readable communication.

The diagram below illustrates a straightforward depiction of the Software Supply Chain Security market in relation to the AppSec, including some of its prominent participants, as well as Scribe's placement.

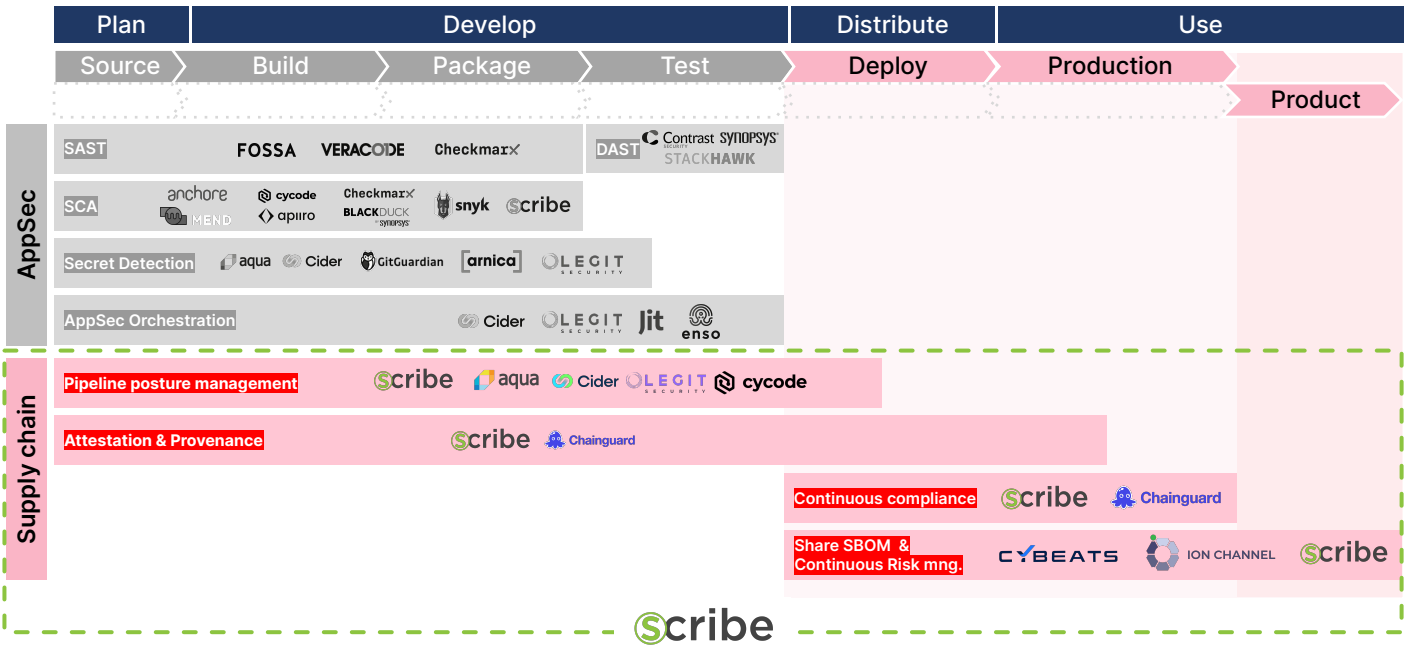


Diagram: Software Supply Chain Security market landscape vs. AppSec

CONCLUSION

As software development and the use of third-party components continue to expand, the need for a comprehensive and integrated approach to software supply chain security has become increasingly important. Traditional AppSec measures are still important but no longer enough to fully protect against the evolving threat landscape, and organizations must adopt a more holistic approach to ensure the security of their software supply chain.

To overcome the evolving security challenges, we are currently witnessing the evolution of Application Security to Software Supply Chain Security. It includes a new generation of technologies and novel tools that try to address these challenges.

Software producers need to integrate new security technologies that take advantage of modern frameworks and tools, and they must do so into all stages of the SDLC and use automated tools, such as CI/CD posture management, attestation and provenance technology, continuous code signing, ongoing compliance, and governance, and accurate, high fidelity SBOMs.

Software consumers should require SBOMs and further evidence-based attestation to all the security aspects of the software they consume on an ongoing basis, before and after the software deployment to production during its entire life cycle.

Automated tools and solutions like Scribe help organizations achieve a new level of security by providing an evidence-based continuous code security assurance platform that can attest to the trustworthiness of the software development life cycle and software components.