REVERSINGLABS

Why Traditional App Sec Testing Fails on Supply Chain Security

Here's how traditional AST tools alone leave your organization exposed to supply chain attacks — and how software supply chain security tools represent an evolution of traditional application security tools, ensuring end-to-end software security.

Traditional application security testing (AST), which includes static application security testing (SAST), dynamic application security testing (DAST) and software composition analysis (SCA), by itself is insufficient to prevent attacks coming from your software supply chain. It leaves your organization vulnerable for three reasons:

- Sophisticated supply chain attacks that target modern development environments are on the rise due to the complexity and speed of modern DevOps processes.
- Software packages become larger and comprise more open-source and third-party code as development progresses.
- Traditional AST approaches rely on testing source code and open-source libraries for vulnerabilities instead of focusing on how the code behaves.

Traditional AST solutions are narrowly focused on source code, vulnerabilities, and open-source licensing compliance. While this approach can identify vulnerability and exposure risk in internally developed code and open-source libraries, it completely overlooks third-party software components.

Traditional AST leaves you and your customers exposed to supply chain threats, including malware, unanticipated behaviors, and secrets compromises. Software supply chain security tools, which provide deep visibility into threats and take a more comprehensive risk approach, can mitigate the threats and exposures end-to-end.

This report explains where the rise of supply chain attacks is coming from, why traditional AST leaves you and your customers exposed to supply chain threats — and how software supply chain security represents a much-needed evolution of application security to address the problem.

A rising tide of supply chain attacks

Two years ago a cyber breach at the U.S. Treasury and Commerce departments[1] enabled attackers to monitor internal email traffic. The cause — malicious additions to network monitoring software distributed by SolarWinds — ultimately affected more than 18,000 organizations worldwide[2]. What rattled security experts most, however, was that the additions were introduced through SolarWinds' approved and digitally signed update channels.

Unfortunately, the incidence of software supply chain attacks has only risen since, with the subsequent IconBurst, Material Tailwind, and CodeCov[3] attacks, to name a few. Because of that, supply chain attacks have become a top-of-mind issue for dev teams[4], according to a recent survey of 300 technology professionals. Existing AST solutions by themselves can't protect your organization from these types or attacks. You need software supply chain security (SSCS) tools to mitigate these threats.

The issue has gained the attention of governments in the U.S., Europe and the U.K., which have issued new supply chain security guidance, as well as requirements in the U.S. for software sold to government agencies.

4 COMMON SUPPLY CHAIN ATTACK VECTORS THAT TRADITIONAL AST MISSES

1

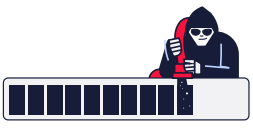
Typosquatting,
which adds malicious code to a legitimate file and changes the file name or version number to something similar to fool developers into using the modified file.

2

Bypassing commit controls.
Attackers distribute malicious code through unofficial and unmonitored channels, promising lower costs or faster download speeds.

3

Software distribution network attacks.
These compromise legitimate software updates - particularly those distributed automatically, by replacing approved code with malware or backdoors.



Where traditional application security testing falls short

There's no shortage of tools for analyzing source code to identify known vulnerabilities (see the figure below), but application security testing software integrity validation tools fail to sufficiently address open-source and third-party software risks that come later in the development process, such as malware, code tampering and compromises in build platforms. Here are where the shortfalls lie.

The software supply chain security puzzle

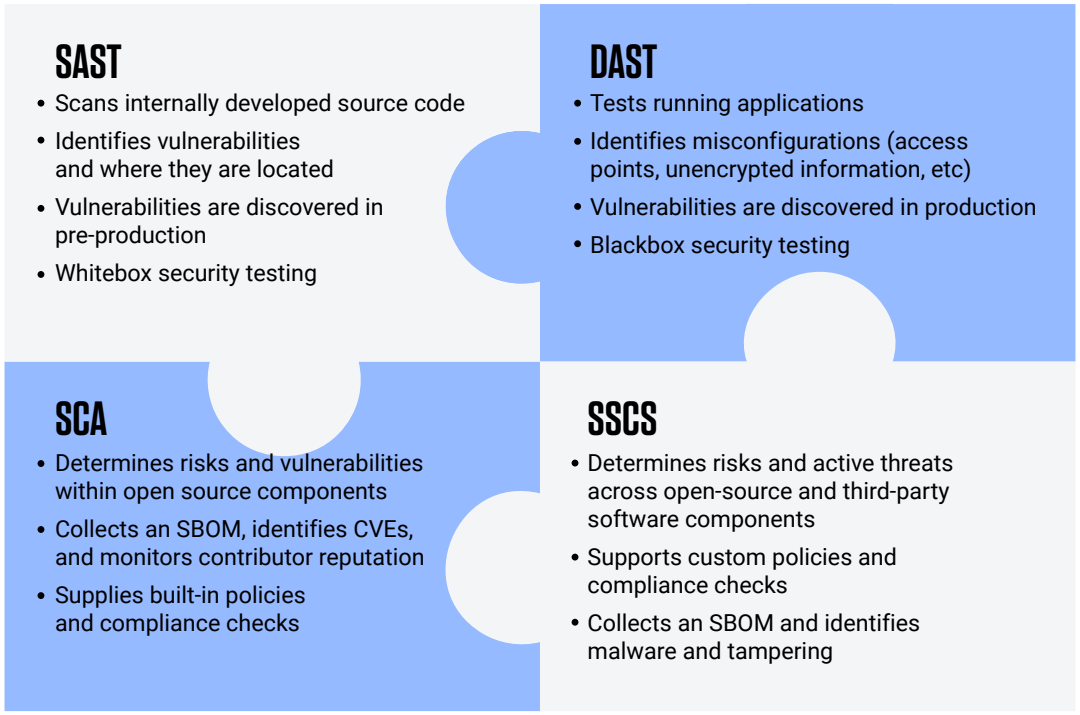


Figure 1. In the software supply chain security puzzle, SSCS is the missing piece that extends the functionality of SAST/DAS and SCA tools. *Source: ReversingLabs.*

SAST and DAST trip up on tampered code, offer only a limited view of software risk

Traditional app sec tools such as SAST and DAST are useful for ferreting out vulnerabilities and reviewing code quality in source code as it's being developed and executed, but they can't identify backdoors, malware, or other malicious elements introduced into the development process via modern software supply chain attacks.

4

Functionality vulnerabilities.

Attackers leverage inherent weaknesses in open-source development that can allow an error to go undetected despite commit controls.

5

CI/CD platform attacks.

The hack of CircleCI's CI/CD platform[5] and associated secrets leak in January 2023 is yet another wake-up call for supply chain security.

> Software supply chain security needs to be recognized for what it has become:
>
> A separate discipline within the application security ecosystem.

SCA only focuses on open-source and misses malicious modules, offers only limited software bill of materials

SCA provides only limited visibility of complete software packages. It uses a limited automated binary analysis approach to identify open-source software in a software package and pinpoint open-source vulnerabilities, but SCA can take an extremely long time to analyze large, complex binaries—while only giving you a limited open-source view of the software package. SCA is valuable for creating a traditional software bill of materials (SBOM), but will miss third-party components and the detection of malicious modules.

SCA provides only a limited software bill of materials (SBOM). SCA is limited to presenting only open-source libraries in an SBOM. It misses third-party components and packages, leaving software supply chain risks open. The lack of comprehensiveness makes operationalizing an SBOM for future package governance problematic.

Software vulnerabilities aren't the whole picture

Most application security testing tools are more oriented toward spotting vulnerabilities than malicious code. This distinction is important because vulnerabilities are usually the result of human error, whereas malware is introduced intentionally. Attackers go to great lengths to cover their tracks and fly under the radar of automated code scanning solutions.

Known, reported vulnerabilities are also just one part of the picture. NIST's National Vulnerability Database, which is considered the most comprehensive database of known vulnerabilities, is dominated by flaws in a few legacy platforms. However, participation and submissions of vulnerabilities are voluntary, not mandatory.

Today, the NVD does not cover the full scope of development tools and platforms that are increasingly being targeted. As a result, flaws in platforms that are widely used, but where the vendor is not a CNA (CVE Numbering Authority) may get overlooked. Also, the focus on software vulnerabilities overlooks other, growing supply chain threats such as malware injections, software tampering and secrets exposure.

ReversingLabs NVD Analysis 2022:
A Call to Action on Software Supply Chain Security

LEARN MORE IN OUR SPECIAL REPORT

The state of AST tools

As critical as AST tools are for securing software supply chains, they are only part of the picture. All too often, organizations equate SCA to software supply chain security. SCA is an important part of software supply chain security, but it's just one facet of it — the facet that deals with open-source packages. Software supply chain security needs to be recognized for what it has become: A separate discipline within the application security ecosystem.

BEYOND OPEN SOURCE: SIX KEY ATTACK TYPES TO WATCH

Open-source repositories aren't the only supply chain attack vector. Others worth mentioning include:

1

COMMERCIAL THIRD-PARTY SOFTWARE

is only as secure as the controls used to develop it. Business users routinely download productivity applications to their computers without verifying that the source can be trusted. A simple redirect exploit can send those download requests to a file containing malware. And many developers of plug-ins and utility software are small shops with questionable security controls.

2

APP STORES.

More than two dozen app stores are available for mobile devices, each with its own security practices. Other publishers forgo the app store and deliver software directly, bypassing any protections an app store might provide.

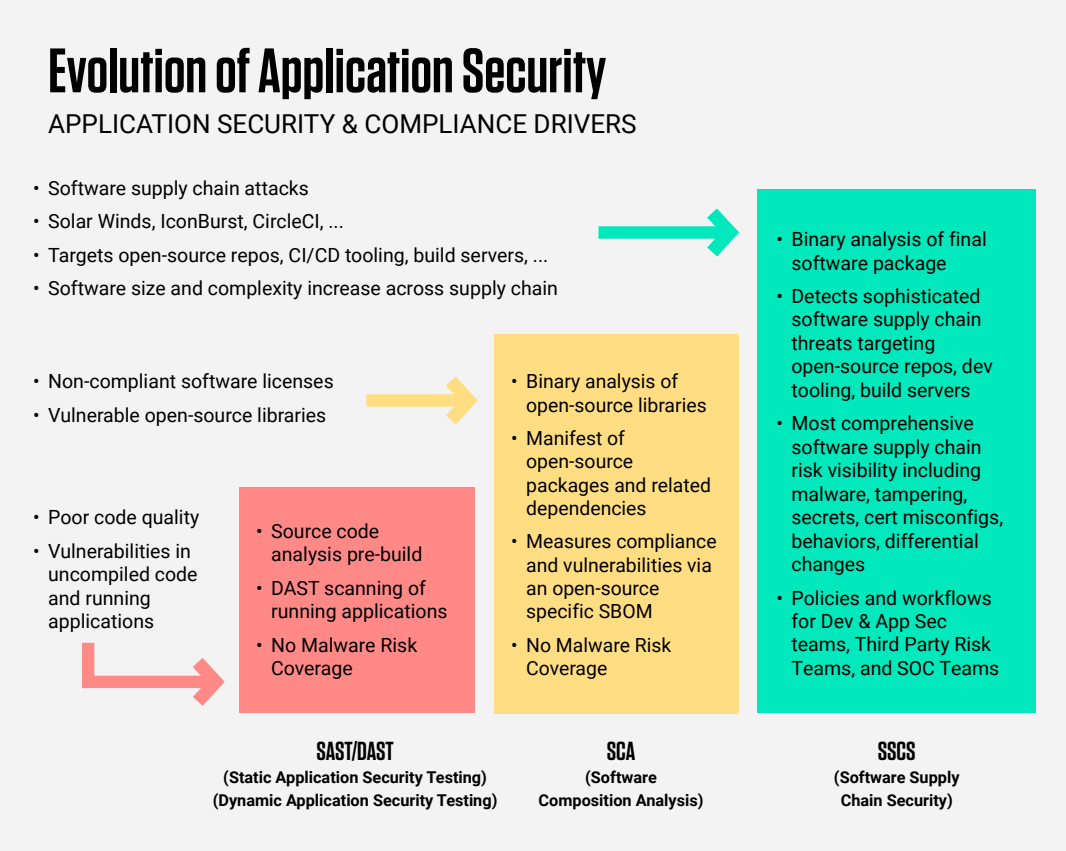


Figure 2. SAST/DAST was designed to address poor code quality and vulnerabilities in uncompiled code and running applications. SCA addresses non-compliant software licenses and vulnerable open-source libraries. SCSS can help protect against software supply chain attacks as software size and complexity continue to increase across the supply chain. *Source: ReversingLabs.*

3

PLATFORM-SPECIFIC CODE REPOSITORIES.

Many code repositories are specific to each platform. For example, NuGet[6] hosts more than 300,000 packages related to the .NET framework. Many are DLL files, which can be shared by multiple programs. A vulnerability in a DLL file can affect many other connected applications.
And because such software is difficult to detect and update once it's in the field, a vulnerability may persist in applications years after a patch was issued.

4

FILELESS MALWARE.

Some attacks don't require source code. Fileless malware[7] injects itself directly into memory after the software has loaded from storage, making it nearly impossible for conventional malware protection software to detect.

5

APPLICATION PROGRAM INTERFACES (API),

which enable businesses to selectively expose data and services, are now one of the most frequent sources of data breaches[8].

6

OTHER ATTACK SOURCES

include the images[9], PDF files[10] and macros[11] distributed within documents, multimedia and file archives included in software packages.

Legacy Gaps and Modern SSCS

	SCA	SAST	DAST	SSCS
Software Bill of Materials	●			●
Binary Analysis	◗	◗		●
Extensive Coverage of Binary Formats				●
Pre-Production Scanning	●	●		●
Production Scanning			●	●
Attack Threat Intelligence				●
Malware & Malicious Behaviors				●
Tampering Detection				●
Version Differencing				●
Digital Signature Validation				●
Secret Leakage Detection	●			●
CVE Detection	●	●	●	●
Contextual Alerting	●		●	●
Custom Policy Enforcement				●
Multi-Team Support: Dev Sec SOC IT Compliance Risk etc.				●

Figure 3. SCA, SAST, DAST and SSCS: Key features compared. *Source: ReversingLabs.*

End-to-end software supply chain security is key

Innovations such as agile development and cloud-native computing have led to breathtaking advances in the speed with which software is built, but they have also uncovered new vulnerabilities and exposed supply chain weaknesses that have existed for many years.

Traditional AST focuses on the needs of development, testing, and application security teams, but provides only a limited view of risk and actionable information for each group. Shift-left software security assurance approaches that focus on vulnerability scanning and remediation during software development and release processes alone aren't sufficient to keep up with software supply chain risks.

The complexity of software supply chain attacks is creating a multi-team problem. Development and application security must ensure that software packages are securely released, third-party risk management teams are tasked with mitigating the risk of commercial software coming into the organization, and the SOC is the last line of defense, requiring more visibility into software supply chain threats to accelerate detection, isolation and response when attacks make their way through layered defenses. All teams need to collaborate. They need to automate the assessment of all software packages prior to release, purchase, deployment and updating, while tracking all software risks for compliance and extending visibility to the right teams to take action.

To accomplish that, these teams require an end-to-end platform that can go beyond the limitations of an application security tool and provide a capability that can democratize decision making across the enterprise. This requires a software supply chain security (SSCS) platform that goes beyond the limitations of SAST source code analysis and SCA open-source binary analysis. Software supply chain security solutions are the next step.

Software Supply Chain and the SOC: End-to-End Security is Key

LEARN MORE IN OUR SPECIAL REPORT

SOURCES:

1 https://www.reuters.com/article/uk-usa-cyber-treasury-exclusive-idUKKBN28N0PI

2 https://www.zdnet.com/article/sec-filings-solarwinds-says-18000-customers-are-impacted-by-recent-hack/

3 https://www.reversinglabs.com/blog/a-partial-history-of-software-supply-chain-attacks

4 https://www.reversinglabs.com/blog/survey-finds-software-supply-chain-security-top-of-mind-for-dev-teams

5 https://www.reversinglabs.com/blog/circleci-hack-is-a-red-flag-for-security-teams-on-the-software-supply-chain

6 https://www.nuget.org/

7 https://learn.microsoft.com/en-us/microsoft-365/security/intelligence/fileless-threats?view=o365-worldwide

8 https://www.gartner.com/en/documents/3834704

9 https://www.reversinglabs.com/blog/malware-in-images

10 https://www.reversinglabs.com/blog/the-pdf-invoice-that-phished-you

11 https://www.reversinglabs.com/blog/spotting-malicious-excel4-macros

REVERSINGLABS SPECIAL REPORTS:

ReversingLabs NVD Analysis 2022: A Call to Action on Software Supply Chain Security:
https://www.secure.software/reports/reversinglabs-nvd-analysis-2022-a-call-to-action-on-software-supply-chain-security

The State of Software Supply Chain Security 2022-23:
https://www.reversinglabs.com/the-state-of-software-supply-chain-security

The Evolution of Application Security:
https://www.reversinglabs.com/the-evolution-of-application-security

Software Supply Chain and the SOC: End-to-End Security is Key:
https://www.reversinglabs.com/reports/software-supply-chain-soc-end-to-end-security-key

ReversingLabs Fills the Gaps

ReversingLabs can help fill these technology gaps by providing a software supply chain security solution with the following critical capabilities:

WORLD'S LARGEST ATTACK INTELLIGENCE REPOSITORY:

Over 14 years of experience aggregating malware / goodware privately based on 46 AV scanners and a threat intel platform that adds 8M+ per day.

PROPRIETARY RECURSIVE BINARY ANALYSIS:

Extensive coverage of binary formats. Unpacks, deobfuscates, extracts metadata, and classifies down to the lowest level. Unpacks more than 400 file formats to create the most comprehensive SBOM. Identifies more than 4,800 file types (JAVA, .NET, Python, Mac OS, Linux, MS Office, PDF, Docker, etc.) for malware detection.

INDUSTRY-LEADING ANALYSIS SPEED OF LARGE, COMPLEX SOFTWARE PACKAGES:

Analyzes the largest proprietary and open-source complex files in seconds – 10GB+ files at 10M files per day – enabling frictionless release and deployment.

COMPREHENSIVE SOFTWARE RISK VISIBILITY AND PRIORITIZATION:

Provides the most comprehensive software risk visibility of malware, tampering, differential behaviors, secrets, certificate misconfigurations and dependencies to prioritize remediation, release, deployment and decision making.

EXTENSIVE POLICY AND SOFTWARE SUPPORT:

Includes detection, prioritization, remediation, validation, and interactive reports and search. Customized policies for different projects, applications, and individual components.

END-TO-END DEVELOPMENT, SOC AND RISK TEAM SUPPORT AND WORKFLOWS:

Democratizes software decision making across teams, enabling development and application security teams to safely release, IT and procurement teams to securely deploy, the SOC to detect, isolate and respond, and risk and audit teams to comply with internal and external standards and mandates.

An end-to-end software supply chain security solution is the next evolution of application security, and is fast on its way to becoming a standard part of the most widely respected cybersecurity frameworks. The time for development, security operations, and third-party risk management teams to embrace that change with ReversingLabs is now.

Book a demo or speak to us about how to close gaps in your Software Supply Chain Security

REQUEST A DEMO

Worldwide Sales:
+1.617.250.7518
sales@reversinglabs.com