Checkmarx

The world runs on code. We secure it.

SCS

Checkmarx Supply Chain Security

Automated Policy-Driven Open Source Security for DevSecOps

A significant challenge of modern application development stems from potential supply chain threats due to the ever-increasing use of open source software (OSS).

Rather than recreating basic functionality to power applications, developers leverage open source code as building blocks, allowing them to concentrate on their own code. While this increases development speed and agility, it also exposes organizations to challenges which amount to taking code from strangers.

Strategies to maintain trust in the use of open source software include:

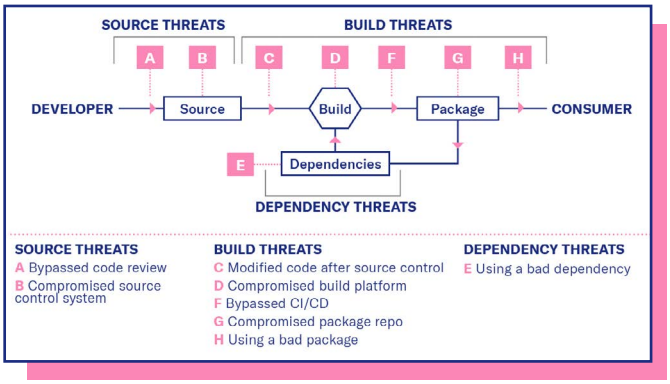
- Knowing which open source packages are included via a Software Bill of Materials (SBOM).
- Detecting malicious packages introduced by techniques such as typosquatting. chainjacking and other anomalous activity
- Evaluating the activity of contributors across all projects.
- Observing the behavior of packages via static and dynamic analysis within a detonation chamber.
- Continuous threat hunting via a threat intelligence database.

What's Not Working

Traditional Software Composition Analysis (SCA) is most often associated with identifying the OSS packages being used. More advanced solutions include basic supply chain anomaly detection. However, many organizations struggle with comprehensive visibility and security of their supply chain.

Although the stages of the development life cycle can be secured by scanning code early and often, hardening build systems, and so-on, organizations have little control over the security of open source packages in use in their code base. They can't possibly monitor or even scan every open source project they utilize.

Of the three trust boundaries identified by the Supply-chain Levels for Software Artifacts (SLSA) framework, (source threats, build threats, and dependency threats) dependency integrity is the weakest link in the software supply chain.



Supply chain Levels for Software Artifacts (SLSA)

An Ecosystem Not Built for Security

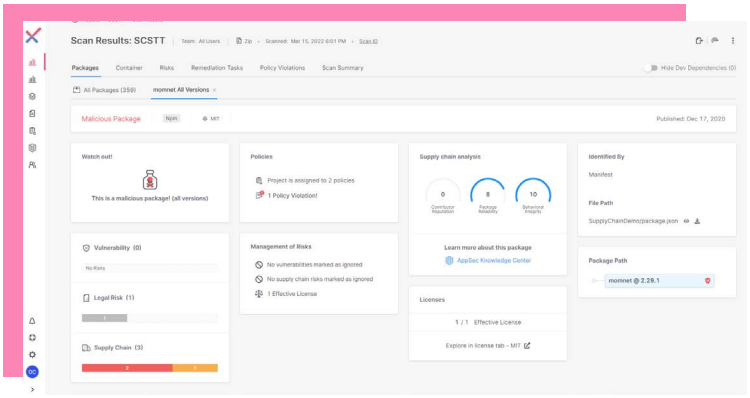
Open source software ecosystems like GitHub and others are built to minimize friction and encourage developers to contribute to the community. As such, the mechanisms for contributors are built to make it easy, and there is inherent trust built in. However, security is not always a top priority. Some projects that are backed by well-known vendors often have checks and balances around who can contribute, what is contributed, what makes it into builds, and so on. However, lots of projects don't have the same level of controls, leaving them susceptible to attacks which, in-turn, can end up in the software stack of an application.

While traditional SCA identifies packages with known vulnerabilities (CVEs) or packages needing updating to a latest revision, supply chain security focuses more on tactics, techniques, and procedures (TTPs) bad actors use to hijack projects and inject purely malicious code into often extremely popular open source packages.

The Solution to Managing Supply Chain Risks

Checkmarx Supply Chain Security (SCS) brings a level of assurance to organizations who must protect their applications from attacks that leverage open source software. Checkmarx uses a variety of constantly-evolving methods to drive the unique value of Checkmarx SCS by addressing

- **Stealthy threats** can be included in the packages which your applications rely on. The sheer volume of packages and updates makes it all but impossible to closely inspect everything without an automated solution. Remove ambiguity with the detonation chamber to provide deep analysis of package behaviors.
- **Tracking everyone and everything** in the open source ecosystem without a sophisticated solution is practically impossible. Ensure trust in the open source elements you leverage with automated reputation analysis of project and contributors.
- **Gauging the health and security** of a project means much more than skimming through the read-me. As with other aspects of supply chain security, the scope and scale of this problem makes it impractical for your teams to achieve. Gain assurance in project integrity with continuous metadata analysis which identifies anomalies in the integrity of a project.
- **Remediate threats** in the earliest stages of your development cycle is imperative. Overwhelming your team with alerts long after code has shipped is not optimal. Instead, enable zero friction between security teams and developers. Checkmarx integration with the IDEs your developers know and love, means they will never have to see a security console. Relevant risk information is available when, where, and how developers want an
- **Require a futureproof approach** since the threat landscape continues to evolve. Checkmarx SCS has been architected such that engines can be updated, or entirely new engines added quickly and easily, means the solution will grow with you, not against you.



Conclusion

When embracing modern application development, your organization is gaining tremendous agility, but the risks of taking code from strangers are very real and growing.

Rather than reactively dealing with malicious code introduced through the open source elements of your software supply chain, shifting security left to get it closer to the code itself, results in a far more proactive approach to security, and effectively managing your risk.

Not only does Checkmarx SCS improve the security of your applications, but it also costs less to remediate risk in the code itself, rather than when an application is in production. No one wants to scramble to find where a problem is and what to fix, in a running application.

Contact us for a free demo ***here***.