Checkmarx | TD SYNNEX Public Sector | DLT

EBOOK

10 Key Considerations

When Choosing a SAST Solution

Essential elements for selecting a SAST solution as part of a comprehensive AppSec strategy

At a glance

Key considerations

With so many different application security solutions on the market, it's tough to sometimes know where to start.

It makes sense to start with SAST. Yet, it can be hard to determine what is the right choice for your organization.

When analyzing vendors, you want to choose a leader. Therefore, it's helpful to explore the Gartner Magic Quadrant™ for Application Security Testing and Forrester Wave for SAST.
Checkmarx is proud to be recognized as a leader by both analyst firms and in both reports.

Here is a quick checklist
to use during your evaluations:

- ☑ Scan deep and wide
- ☑ Target with presets
- ☑ Get the whole picture with application-centricity
- ☑ Correlate and use ASPM
- ☑ Look beyond scan speed
- ☑ Fix problems fast
- ☑ Turn developers into security heroes
- ☑ Support lots of languages and frameworks
- ☑ Integrate and automate
- ☑ Scale security with an AppSec platform

☑ **BONUS:** Measure AppSec maturity

Contents

Attackers Adapt Rapidly; Your SAST Must, Too

Imagine a SAST solution that is tailored to fit your organization. It should handle all sorts of different projects, from internal tools to customer-facing applications. Ideally, it would keep everyone happy – from CISOs to the folks who write the code and the security champions in between (application security teams).

Plus, it should also seamlessly integrate with how your developers like to work and deliver their code. That kind of flexibility is key! Let's dive into the first two points.

01

Scan Deep and Wide

Different teams have different needs. AppSec teams handling mission-critical applications need deep scans. These scans thoroughly uncover vulnerabilities, providing a comprehensive security risk view. However, they require significant effort to analyze and prioritize remediation.

In other cases, AppSec teams with many non-mission critical applications may prefer wide scans. These scans focus on finding only the most critical high-severity vulnerabilities, allowing developers to address the most pressing issues first.

For example, Checkmarx SAST offers the ability to perform in-depth scans for comprehensive coverage on critical applications with a zero-vulnerability policy. It also has Fast Scan mode, which as the name suggests, reduces the scanning time of projects. This allows for quicker identification of relevant vulnerabilities and facilitation of continuous deployments, while ensuring that security standards are followed.

This dual approach allows developers to react swiftly to immediate issues, with Fast Scan mode finding the most significant vulnerabilities and in-depth scan mode providing deeper, more thorough security coverage.

Attackers evolve quickly. This flexibility allows organizations to standardize on a single platform that maximizes security.

02

Use Presets to Target Your Searches

Presets are predefined groups of scan rules for various needs. For instance, you might use a preset for regulatory compliance, or one based on the type of code being scanned.

Organizations need SAST solutions with diverse presets for regulatory compliance, security standards (like OWASP Top 10), mobile and web application development, and more. Some presets scan the entire codebase, while others focus on critical vulnerabilities.

AppSec teams can reduce false positives by using the correct profile for the code, selecting rules for the appropriate coding language, and building custom queries as needed.

The Checkmarx base preset targets the highest priority vulnerabilities, cutting medium and low severity vulnerabilities and reducing the noise by up to 90%. It was designed to boost scanning efficiency, prioritizing the swift retrieval of results with pertinent and impactful vulnerabilities. The preset can also be used as a starting point and customized to meet your specific requirements.

Checkmarx SAST also allows users to customize queries. Even if you're unfamiliar with query building, our built-in AI Query Builder can assist. For example, if your organization has a custom sanitizer, adjusting the queries to recognize it can reduce false positives.

RECOMMENDATION

Use a SAST tool with customizable presets and queries.

Accuracy Matters

Accuracy is crucial. But how much accuracy do you need? Do you need to examine every line of code in detail, or is a broader view enough? Like in the real world, sometimes you need a microscope, other times a telescope, and sometimes just a pair of eyeglasses. Different tools serve different needs. Finding the right balance is critical.

In application security, we often focus on false positives. But false positives alone don't define accuracy. Accuracy means maximizing true positives and true negatives, while minimizing false positives and false negatives. In other words, how do you find the most real vulnerabilities with the fewest false positives? Different applications may require a different balance.

03

Get the Whole Picture With Application-Centricity

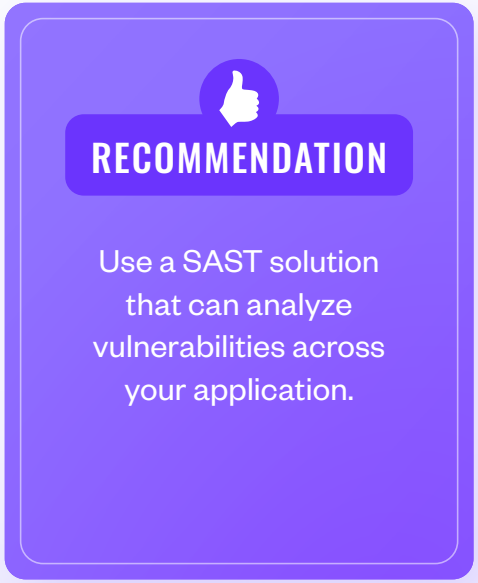
Data flows across multiple files in an application can create vulnerabilities. SAST solutions need to be able to detect them. While this can lead to longer scans, it makes the code more secure by uncovering hidden vulnerabilities and ensuring comprehensive coverage. SAST solutions that use the following methods are more accurate:

1. **Data-flow analysis:** Tracks data flow through the code to identify how data is used and manipulated, uncovering complex security vulnerabilities.
2. **Symbolic execution:** Executes the code with symbolic inputs, exploring all possible paths to find exploitable vulnerabilities.

SAST solutions using both data-flow analysis and symbolic execution, combined with an efficient query language, are much more precise than regex-based tools. Regex-based tools rely on pattern matching and may miss complex vulnerabilities, sacrificing accuracy for speed and thereby increasing risk.

Understanding how application flows interact between files and components is crucial for identifying attack vectors. This approach is also useful for providing the best fix locations, helping developers quickly correct coding errors.

Choose a SAST solution that can go as deep as needed to find vulnerabilities that "good enough" solutions can't.



04

Correlate and Use ASPM

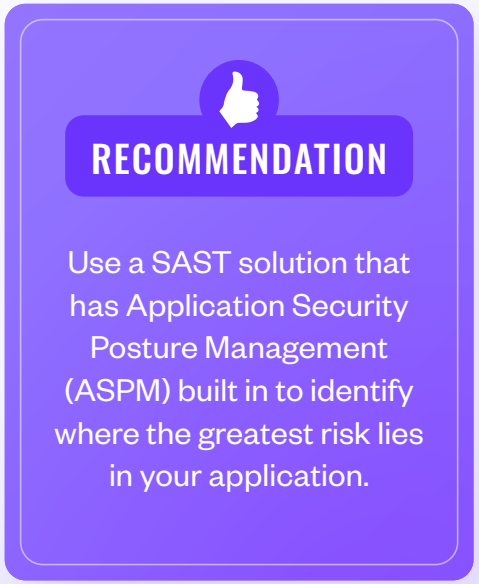
Application Security Posture Management (ASPM) serves as a single source of truth for identifying, correlating, and prioritizing security vulnerabilities across the SDLC.

An ASPM that isn't part of your main AppSec platform but works with a bunch of third-party tools sounds great, but it misses the additional context that those tools find.

You want an ASPM tool built in to your main application security platform – so it leverages your platform's distinctive features and then adds the additional context from third parties.

As a result, you can identify your riskiest applications, understand where the greatest risk lies, and know what to fix first.

By correlating your entire Application Security Testing (AST) environment, you can integrate results from tools like DAST. For example, adding Checkmarx DAST to SAST allows for easier correlation of DAST and SAST vulnerabilities, making prioritization more accurate and straightforward.



The Need for Speed

Development approaches like agile and DevSecOps are the new normal. Security at speed is imperative for meeting deadlines.

In this context, scan speed and fixing problems fast should be considered as top considerations for SAST solutions. Both concepts are discussed in the next section.

05

It's Not Just Scan Speed

Speed is not solely measured by what can scan the fastest. Instead, it is measured by the overall time it takes to accurately scan code AND remediate security issues. Secure software is the goal. It is critical to look at both the time to write code and deploy code.

The way to do that is to encourage frequent scanning.

If you need to compile code each time you want to check, it's simple: you won't check your code as often, and therefore will fix less vulnerabilities.

Select a SAST solution that supports incremental scans, only scanning changed code, and does not require a complete build to launch a scan. Waiting for code to compile before scanning is cumbersome.

SAST solutions should also scan directly from source code repositories like GitHub, GitLab, Azure, and Bitbucket. Scanning at the repository level improves the developer experience and builds #DevSecTrust. With Checkmarx SAST, you can scan automatically on pull or push and decorate pull requests.

What's even better? A tool that provides instant feedback as developers code in the IDE. This reduces vulnerabilities in committed code.

06

Fix Problems Fast

Although application security testing aims to find errors that could lead to vulnerabilities, the ultimate goal is to produce more secure applications. SAST solutions must guide developers to the biggest risks in their applications and advise them on the best remediation approach, reducing Mean Time to Remediation (MTTR).

One way to quickly fix errors is through auto-remediation, which provides the exact code snippet—not just sample code—to fix a specific vulnerability. This means that developers don't need to be security experts to fix vulnerabilities. This also enables developers with low security knowledge to fix it fast, without requiring help from more senior developers.

Some SAST solutions analyze code deeply to understand its context throughout the entire application. This enables them to identify not only coding errors but also the best fix location. By changing code at the best fix location, a single fix can address multiple vulnerabilities, reducing the number of necessary code corrections. Even better, our in-IDE real-time scanning helps scan code as developers are writing it, meaning vulnerabilities never get checked in to begin with!

Organizations aiming to reduce MTTR should choose SAST solutions with deep code understanding and auto-remediation. These solutions guide developers to the precise error location, often resolving multiple issues with one change. This reduces the total number of issues and simplifies triage, speeding up remediation.

RECOMMENDATION

Use a SAST solution that can identify the best location to fix multiple vulnerabilities at once. Even better if it can automatically provide the fix with auto-remediation.

Factor in What Makes up a Developer-Friendly Solution

Developers build the applications that your organization runs on. But the entire software development life cycle (SDLC) is made up of many moving parts: planning, coding, testing, integration, deployment, packaging, and more. Developers don't want to have to learn another new tool, change the way they develop and deploy software, have another tool interrupting their workflow, or add more steps to the SDLC.

Choose a SAST solution that helps build, what we at Checkmarx call #DevSecTrust. Build trust between security and developers by prioritizing for the greatest impact, meeting developers where they live, and equipping them with the tools and knowledge they need.

SAST solutions must be designed with developers in mind and fit well into their daily activities. Developer adoption of SAST solutions is crucial for enhancing security.

07

Turn Developers Into Security Heroes

Developers are not security experts. That's not their job. By equipping them with a SAST solution that shows how to fix vulnerabilities and educates them about secure coding, they can write more secure code, making their jobs easier.

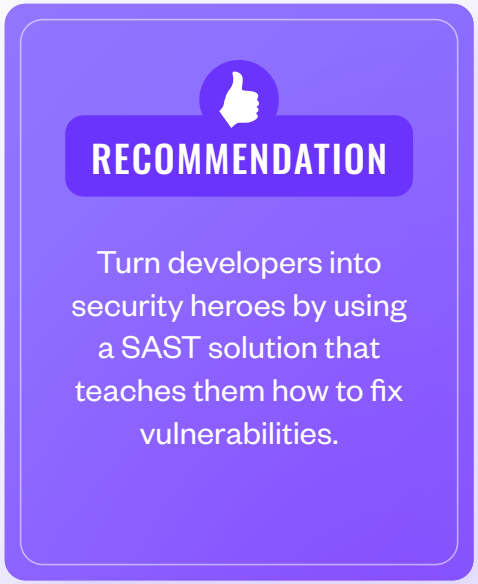
Some SAST solutions leave developers in the dark. Other SAST solutions, like ours, prevent vulnerable code from being committed with real-time in-IDE guidance. This immediate feedback ensures that only well-written, secure code is committed.

If vulnerabilities slip through, these solutions (like ours) clearly pinpoint where and how to fix them, offering remediation guidance and even replacement code to auto-remediate issues with a single click.

Some solutions also deliver comprehensive secure coding training modules, designed to increase developers' skills and security awareness, such as Codebashing. Bite-sized, gamified training increases developer adoption and may enhance employee retention.

With the right SAST solution, your developers likely won't need to go to Stack Overflow or Reddit seeking advice on how to fix an issue.

The ultimate training to enhance security-minded skills combines the ability to get trained, obtain remediation advice, and edit code right from the tools developers are using. SAST solutions that incorporate training into the process of writing code ultimately reduce time spent on fixing code.



Secure Code Makes Happy Developers

The Need for Ecosystem Support

No two software development environments are the same. This is where the broad spectrum of languages and frameworks must be addressed, and where integration and automation of SAST scans are imperative.

08

Support Comprehensive Languages and Frameworks

Choosing a programming language or framework often depends on preferences, requirements, and organizational standards. For instance, C++ is common in computer games and embedded systems, while Swift and Dart are popular for mobile app development.

Developers prioritize their goals and deadlines, and certain languages excel in specific tasks. Recognizing this, SAST solutions must inherently support a wide range of languages and frameworks.

For example, if you're in finance, the organization may need to support both legacy languages such as COBOL, which still powers banking transactions, in addition to the latest mobile development languages such as Flutter and Dart. What your developers are using today aren't necessarily what they are using tomorrow. It's important to use a solution that constantly innovates and adapts to changing market conditions, so you don't have to rip it out and replace it with a new solution if you start developing in a new language.

Organizations can maximize efficiencies by standardizing on a single application security platform, rather than resort to a mishmash of vendors.

When selecting a SAST solution, look for vendors who not only support the largest number of languages and frameworks, but also how often they add new languages that are trending in the industry, so that you can future-proof your application security platform.

RECOMMENDATION

Use a SAST solution that supports a wide array of languages and frameworks.

09

Don’t Forget Integration and Automation

The software development lifecycle is intricate, with numerous stages and complexities. Introducing interruptions or additional steps can frustrate developers, delay secure deployment, and discourage tool adoption within your organization.

Integrating and automating SAST solutions throughout the development lifecycle is crucial for fostering widespread adoption and achieving comprehensive security practices.

Here are examples of key integrations:

- Source Code Management (SCM) solutions (Bitbucket, GitHub, GitLab, etc.)
- Integrated Development Environment (IDE) solutions (Eclipse, IntelliJ, Visual Studio, etc.)
- Continuous Integration/Continuous Delivery (CI/CD) solutions (Jenkins, CircleCI, Bamboo, Team City, etc.)
- Feedback solutions (Azure DevOps, Jira, Rally, etc.)

Custom integrations are also important since organizations may use less common tools. The goal is to seamlessly embed security testing into development, making it a natural and accepted part of the process without causing delays.

By integrating SAST into everyday processes, developers and release managers receive early warnings about vulnerabilities. When executed effectively, fully integrated and automated security testing accelerates secure application delivery and deployment.

It's essential for application security testing to align seamlessly with existing workflows, reducing friction, accelerating adoption among developers, and advancing secure software initiatives. When evaluating SAST solutions, prioritize their ability to integrate and automate effortlessly within your specific development environments and processes.

RECOMMENDATION

Use a SAST solution that integrates and automates into development.

Security Must be Expandable and Scalable

To produce more secure software, organizations need multiple application security testing (AST) solutions.

These tools address vulnerabilities in different types of code within a modern application. Beyond SAST, organizations often use DAST, SCA, Software Supply Chain Security, Container Security, and API Security.

Unifying these tools in a single AppSec platform offers a comprehensive view of risks and vulnerabilities and allows for correlating results across scanning engines for more accurate outcomes.

10

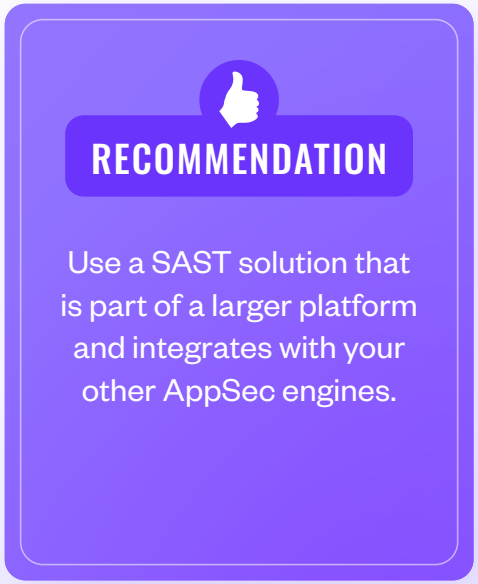
How a Security Platform is Essential

There is no need to piece together multiple point products to secure modern applications. Relying on isolated solutions leads to complex integrations and unnecessary costs. Instead, a platform-based approach integrates various Application Security Testing (AST) capabilities into a single unified platform.

Designed for today's technology stack, processes, vulnerabilities, and risks, a comprehensive application security platform simplifies security for applicative code, open source dependencies, supply chains, IaC, APIs, containers, and more—all from a single scan. Built from industry-leading AST solutions, such a platform provides rapid, correlated, and accurate results to speed up remediation.

Ensure the platform can correlate scan results across different engines to provide a holistic risk assessment across projects and applications, eliminating the need for manual aggregation from standalone AST tools.

When choosing a SAST solution, opt for one that is part of a unified platform for the best value in securing modern applications. A complete platform should offer a centralized dashboard for SAST, SCA, SSCS, API security, DAST, IaC security, and container security. It should also be able to grow with your needs over time. It shouldn't be stitched together from acquired products but rather – like Checkmarx One – built from the ground up.



Bonus

AppSec Maturity Modeling Can Boost Your ROI

Even the best tool may not elevate your organization to the next level in their application security maturity. Guidance from application security and change management experts can be pivotal in achieving this advancement.

Organizations can derive significant benefits from professional methodologies aimed at optimizing the use of AST solutions and fostering adoption. An AppSec maturity model is instrumental in maximizing value and achieving rapid return on investment. Key advantages of adopting a maturity model include:

- **Rapid assessment and understanding**: Quickly assess and comprehend the current state of application security activities.
- **Identifying gaps and tracking progress:** Identify gaps in security practices and measure progress over time through periodic assessments.
- **Strategic planning and execution:** Develop a clear vision of the desired end-state and execute targeted steps to achieve security goals.
- **Best practices guidance:** Access best-practice guidelines for program components to align internal stakeholders and enhance security capabilities.

Implementing an AppSec maturity model empowers organizations to systematically enhance their security posture, drive efficiency, and effectively manage risks across their application landscape.

Checkmarx SAST

The Cornerstone of Comprehensive AppSec

Application security demands a top-tier SAST solution, and Checkmarx has been a pioneer in this field for nearly two decades. Recognized by analysts as a leader in cloud-native application security, Checkmarx stands out by offering flexibility and robust security solutions that cater to the entire organization: developers, application security teams, CIOs, and CISOs alike.

Checkmarx SAST seamlessly integrates into the Checkmarx One platform, providing scalability, correlation of data, and a unified user experience. It empowers teams to prioritize threats, focus on critical fixes, and pinpoint the riskiest applications effectively.

The Checkmarx One AppSec platform provides everything you need to secure your application development. You can take advantage of the capabilities you're ready to use today, while building on a platform with capabilities you know you'll need down the road.

FORRESTER® WAVE LEADER 2023 Static Application Security Testing

Gartner Peer Insights Customers' Choice 2023

Gartner® A leader in the 2023 Garter Magic Quadrant™ for Application Security Testing

TD SYNNEX Public Sector | DLT

Discover how Checkmarx SAST can help you secure your apps, now.

Request a Demo ↗

Checkmarx

Checkmarx is the leader in application security and ensures that enterprises worldwide can secure their application development from code to cloud. Our consolidated platform and services address the needs of enterprises by improving security and reducing TCO, while simultaneously building trust between AppSec, developers, and CISOs. At Checkmarx, we believe it's not just about finding risk, but remediating it across the entire application footprint and software supply chain with one seamless process for all relevant stakeholders.

We are honored to serve more than 1,800 customers, which includes 40 percent of all Fortune 100 companies including Siemens, Airbus, Salesforce, Stellantis, Adidas, Walmart and Sanofi.