Checkmarx

10

Critical Capabilities for AST in DevOps

A List of Key Solution Requirements That Gets Secure Software to Market Faster

Introduction

Security represents one of the biggest challenges to software development since it often directly opposes deployment velocity. Today, security must be inseparable from DevOps in every imaginable way and organizations need the tools, techniques, solutions, and approaches to help them achieve that objective. As a result, organizations adopting DevOps must deeply consider what critical capabilities for application security testing (AST) are essential to getting secure software to market faster.

Not all AST solutions were made for DevOps agility, and actually, some encumber its primary purpose—speed and time to market. Therefore, organizations are at a crossroads whereby they must make an important decision; either adjust their DevOps initiatives to limp along with the current AST solutions they have in place, or implement new solutions that are fundamentally designed to operate better within this new paradigm.

In this eBook, we'll look closely at the 10 Critical Capabilities for AST that can help organizations accelerate their DevOps initiatives, and we'll provide recommendations on what to consider when comparing various AST solutions. Equipped with this eBook, developer, AppSec, and management teams should be able to make better decisions pertaining to their AST purchases and implementation efforts in the context of DevOps.

#1 Automated AST is Required

Checkmarx addresses AST automation head-on by integrating and automating **CxSAST** and **CxSCA** solutions directly into the existing tools in use so that secure software development can be achieved without requiring anything extra. Let's delve a little deeper into what Checkmarx automation is all about.

Traditionally, AST solutions operate within the CI tooling in use and security scans are normally performed after a merge/build has taken place. This required teams to add AST scan jobs before software could be deployed, and as a result, created unwanted delays. On the other hand, Checkmarx allows organizations to shift AST scan functionality as far left as possible since the automation capabilities act as an orchestration layer that simplifies the implementation and automation of AST in today's modern development environments.

This automation supports existing AST best practices in which scans are launched from CI tools (e.g., Jenkins, Bamboo, TeamCity, etc.) while facilitating greater security testing standards with integration back to SCM and code repositories (e.g., GitHub, BitBucket, Azure DevOps, etc.) This enables fully automated application scanning and delivery of actionable results directly to developers. See Figure 1 for a short list of supported integrations.

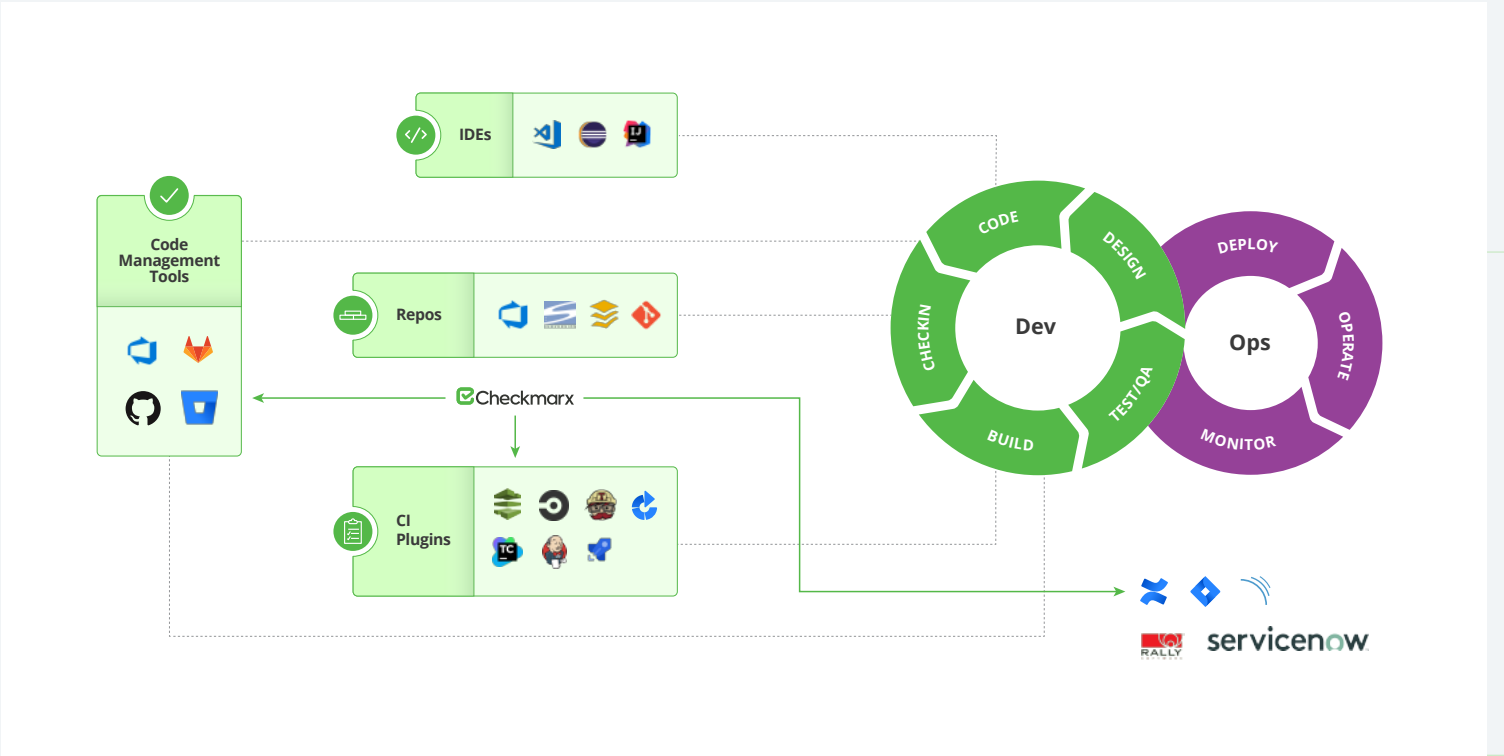


Figure 1

As seen in Figure 1, not only does Checkmarx AST integrate directly with SCM and CI tools, but Checkmarx also performs the necessary integration directly with bug/defect tracking tools like Jira, Rally Software, ServiceNow, SonarQube, etc.

Leveraging a unique ability to scan uncompiled code, Checkmarx AST automates the steps required to scan code earlier in the SDLC. This removes the need for time-consuming manual configuration of scans and allows developers to publish and update scan findings based on pre-configured policies within the code management tools themselves. After the initial configuration, security scans can occur with no human intervention required whatsoever beyond a pull request initiated by a developer.

With Checkmarx AST, a pull request to the SCM tools in use not only initiates an automated scan, but the tools can also receive notifications from scans as comments that are related to sotware functionality *and security.* Just like that, the developer is able to have their code reviewed once for all bugs and would be able to close the full feedback loop with ticketing systems—all while the code is still fresh in their mind. This allows developers to:

- ✓ Catch and fix vulnerabilities during the coding phase (earliest stage of development).
- ✓ Work as usual with no disruptions, no new tools, no additional security reviews, etc.
- ✓ Treat security bugs and functional bugs alike and to immediately address issues within the code branch they're currently working on.
- ✓ Reduce the overhead of manually opening, validating, and closing security tickets with no need to spend countless hours in bug tracking/ticketing management systems.

Checkmarx allows developers to continue working with the tools and processes they are most accustomed to while still raising the standard for software security. Checkmarx also allows developers to fix their own security bugs in the branch of code they're currently working on, locally within the tools they use, before a merge take place and before code moves to security teams for review or into a production environment.

Recommendation

When you begin to compare AST solutions that fit best within your DevOps initiatives, think about the way your development and CI/CD processes are currently being automated in your organization today. Then, assess the AST automation approaches from different venders and their prescribed methods of overlaying their AST technologies on top of your existing processes, instead of changing your existing processes to account for their technologies.

#2 Incremental Scans

Those who have worked directly with SAST solutions often report that scans can take hours to complete. When introducing SAST into a CI pipeline, there might be pushback if full scans introduce delays and software cannot make its way into production quickly. As a result, organizations often reduce the number of full scans taking place, or they run full scans on a nightly basis to account for the expected delays invoked by the scans. The real key to increasing scan velocity is to introduce a SAST solution that has incremental scanning capabilities and is designed to only look at the code changes that were submitted since the last scan.

When CxSAST is automated within SCM tools, scans are launched incrementally against the branch of code a developer is working on. In this way, Checkmarx can provide a protected branching strategy whereby an organization can configure master, development, and security branches, for example, that are all deemed protected. This means that pull requests, push events, etc., will trigger an incremental CxSAST scan and produce results when any code changes are made that are associated with those protected branches.

Recommendation

Organizations trying to retrofit their current AST solutions into their DevOps initiatives often realize that SAST solutions that only support full scans are something they cannot tolerate due to inevitable delays. Knowing this challenge, the recommendation is to consider SAST solutions based upon their ability to incrementally scan uncompiled source code as early and automatically as possible.

#3 Out-of-the-Box Accuracy

Out-of-the-box accuracy is a project Checkmarx embarked on with the goal of delivering highly accurate results via easily consumable rulesets, named Ruleset Content Packs. These content packs are built on Checkmarx's extensive industry knowledge and each content pack's accuracy is verified by multiple open source projects. Each content pack includes improvements to queries and, optionally, to presets. Below are two highlights of Ruleset Content Packs:

- **Lightweight Delivery Model:** Content packs allow Checkmarx to offer frequent and continuous accuracy improvements to CxSAST rulesets and presets without being dependent on the version release schedule. As soon as there are new improvements, Checkmarx releases a content pack that can be immediately consumed by their customers.
- **Continuous Improvements:** Out-of-the-box accuracy is not a one-and-done project. Checkmarx is continuously working on improving queries based on real-world field observations and customer feedback. Figure 2 highlights the continuous process Checkmarx follows:

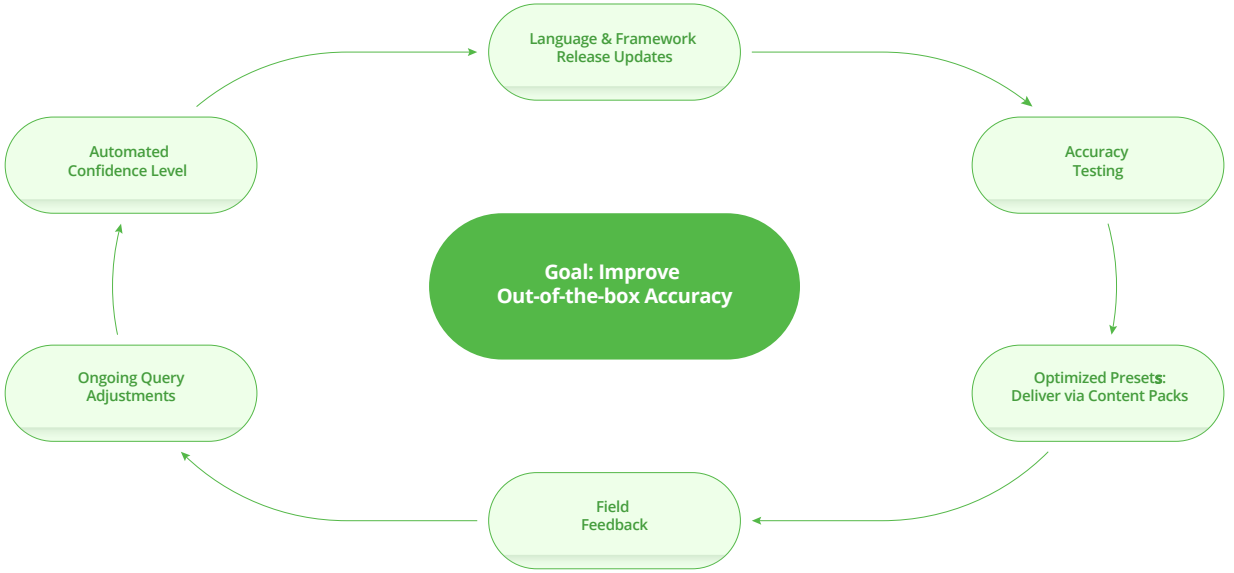


Figure 2

Recommendation

When comparing SAST vendors, make sure their solution supports continuous out-of-the-box accuracy improvements. Ensure that your SAST selection can be updated with mechanisms that improve accuracy without having to upgrade to the latest version. Also, certify that your selected vendor provides incremental improvements to queries and presets since they are of utmost importance to reduce false positives and streamlining your remediation efforts.

#4 Importance of Best Fix Location

When you're building your AST architecture, ensure developers fully understand where vulnerabilities have gotten introduced into their software and how AST technologies fit in to solve that dilemma. More importantly, understanding and demonstrating where the Best Fix Location of a vulnerability is by line number, once they are detected, is of greatest importance.

Figure 3 shows a CxSAST scan summary that tracks OWASP Top 10 2017 for Web Application Risks. Note on the far-right column that the Best Fix Locations are also identified. This information will streamline vulnerability remediation efforts in accordance with each risk category (as shown on the far-left column). Knowing where the Best Fix Location is provides tremendous value to developers and security teams during remediation.

Recommendation

Security scans shouldn't surprise developers when reports indicate their code is full of security bugs. However, security scan reports that provide little, if any, guidance on where or how to fix vulnerabilities tend to frustrate developers. Instead, insist that scans provide relevant guidance on where the best fix location is within the many lines of code. This not only will help developers fix a single bug, but in most cases, when they fix a bug in the right location, other bugs found in latter lines of code may also be remedied.

Category	Threat Agent	Exploitability	Weakness Prevelance	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection*	App Specific	EASY	COMMON	EASY	SEVERE	App Specific	83	51
A2-Broken Authentication	App Specific	EASY	COMMON	AVERAGE	SEVERE	App Specific	0	0
A3-Sensitive Date Exposure*	App Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App Specific	3	3
A4-XML External Entities (XXE)	App Specific	AVERAGE	COMMON	EASY	SEVERE	App Specific	0	0
A5-Broken Access Control*	App Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App Specific	13	3
A6-Security Misconfiguration*	App Specific	EASY	WIDESPREAD	EASY	MODERATE	App Specific	73	73
A7-Cross-Site Scripting (XSS)	App Specific	EASY	WIDESPREAD	EASY	MODERATE	App Specific	10	4
A8-Insecure Deserialization	App Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App Specific	0	0
A9-Using Components with Known Vulnerabilities*	App Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App Specific	0	0
A10-Insufficient Logging & Monitoring	App Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App Specific	0	0

Figure 3

#5 IAST During Functional Testing

Today's software is complicated, often containing millions of lines of code, and to fully test software with DAST can take hours, if not days to complete. With DevOps and CI/CD, there's not a lot of time to wait for DAST testing when releasing two, three, four, or more times a day is the often the norm.

One available alternative to DAST is to augment your existing security testing processes with a solution that finds software vulnerabilities *during functional testing* known as Interactive Application Security Testing (IAST). Organizations are already performing functional testing of their software prior to deployment, and there's no reason not to take advantage of this same process to provide a stream of security data delivered by an IAST solution.

Organizations embarking on DevOps are moving from DAST to IAST because it can be used concurrently within development and functional testing. IAST is clearer and more concise in terms of reporting because it can identify vulnerabilities from inside the application, see the security risk, and watch data flows from entry point to execution at runtime as part of that process.

Recommendation

Since DAST works by attacking the application from the outside, the real issue is that DAST can cause significant delays, and the test results are only as good as the tools in use and the people running them. When considering AST solutions, determine if it's the right time to replace your DAST with IAST, since it does not add any additional delays and it fits well within DevOps initiatives.

#6 Integrated Software Composition Analysis

The practice of using open source components, libraries, and packages during software development allows organizations to accelerate time to delivery, but it can also expose organizations to heightened levels of security risk. For example, organizations that use open source are exposed to new risks that materialize as a result of attackers taking advantage of widespread usage and the public nature of open source.

In addition, organizations are exposed to license risk, since open source components are governed by licenses (e.g., GPL, Apache, etc.) that set terms for the use of the components. And finally, organizations are exposed to operational risk (e.g., technical debt) because the open source support model depends on a community of contributors. Unfortunately, a community can abandon a particular component, version, or fork, and then the organizations using it in their software are left to patch it or evolve it on their own.

CxSCA is the perfect solution for organizations who need to address open source within their DevOps initiatives by detecting and identifying open source components within their code base and providing detailed risk metrics regarding vulnerabilities, potential license conflicts, and outdated libraries. Integrated directly with CxSAST, CxSCA enables development and security teams to prioritize security risks and focus remediation efforts where they will be most effective and least costly.

Recommendation

When considering AST solutions as a whole, ensure that SCA technologies you investigate are fully capable of integrating with your SAST solution of choice, while also integrating into developer tooling already in use. SCA solutions must have the ability to launch scans from code management platforms and from CI solutions already in place.

#7 Exploitable Path, Dev Dependencies, and Dependency Path

CxSCA leverages CxSAST to analyze the structure of the application, visualize the presence of a vulnerable open source component in the execution path of the application, and helps developers and security teams understand the conditions necessary for exploitation. This ensures organizations have the ability to prioritize remediation efforts more effectively, reducing the distraction of nonissues, and reducing time-to-remediation. This is known as Checkmarx's Exploitable Path capability.

In some scenarios, developers may use components during development, but those components are not actually present in the application that goes into production. Such a scenario means that any vulnerable components detected which are used during development, but not in production, may not need to be prioritized for remediation, since they don't increase risk in the production application. This is known as Checkmarx's Dev Dependency capability.

The way Dev Dependency works is that after a build stage, an application is compiled into a binary and all dependencies are resolved into the application.

Some dependencies are considered direct dependencies:

- ✓ dependencies declared by the developer and intentionally resolved into the application,

while others are considered transitive dependencies:

- ✓ dependencies resolved into the application by way of being a dependency of a dependency.

CxSCA identifies both direct and transitive dependencies and provides visualization of the dependency structure to clearly show the reason (or source) for the vulnerable component's presence within the software. This can help developers make more educated decisions during development and remediation by determining the most efficient way to address the risk posed by the dependency. This is known as Checkmarx's Dependency Path capability.

Recommendation

When considering SCA solutions, ensure the vendors you're researching support the concepts of Exploitable Path, Dev Dependency, and Dependency Path-like functionalities to help developers and security teams get a clear, visual understanding of the scan findings, ensuring they're able to take steps to remediate or document that they're not at risk.

#8 Solutions Are Needed for Your AppSec Awareness Initiatives

AppSec awareness programs should keep developers up to date on general AppSec news, security announcements, and specific training activities. Examples include: a weekly security best practice tip, a monthly training reminder, a quarterly security challenge, and an annual company secure development guideline. In addition, AppSec awareness programs can gain tremendous benefit from a centralized platform where managers can:

- ✓ **Enroll** developers and track their course assignments and progress.
- ✓ **Create** teams for friendly competition and tournaments between developers.
- ✓ **Add** training periods, lesson points, challenge points, and earned points for completed courses.
- ✓ **Track** engagement summaries, training progress, most active users, and manage new content.
- ✓ **Maintain** full control, visibility, reporting, and continuously track developers' progress.

Recommendation

Look for AST solutions that include an AppSec awareness platform that helps security teams raise the AppSec knowledge baseline across the entire development team in a fast, scalable, and positive manner. The philosophy behind the right solution is to empower developers in the long term by teaching them how to think and act with a secure mindset, rather than how to only solve specific issues.

#9 Innovative Security Training Works Best

Structure your developer training efforts so that it's delivered in smaller bits over longer periods of time. Fortunately, a training method known as gamification exists, enabling organizations to deliver this information in a more engaging, interactive, and motivating manner. This way, your developers will get the reassurance and retraining they need in a way they can absorb and use immediately—while they're coding. Additionally, by being proactive, you'll reduce risk in your organization and alleviate repetitive coding errors. These proactive steps should entail:

- ✓ **Train, train, and retrain:** Conduct focused training periodically, on a regular schedule, targeting the most common coding errors your organization observes.
- ✓ **Create a friendly competition among developers:** Since everyone enjoys some level of competition, add it to the training mix, and make sure no one feels left behind or left out.
- ✓ **Add an incentive program:** Since most people are motivated by awards and kudos of some kind, think of incentives your teams would appreciate.
- ✓ **Assess improvement:** Don't forget to continuously assess your program and your teams' improvements while keeping good records of all progress and problem areas to target.
- ✓ **Address problems head-on:** If problem areas are identified, address those areas first, and then share positive progress with the group and with a larger audience.
- ✓ **Share the progress:** Run periodic reports that can be easily digested and share that information with management so they understand the value of developer training.

Recommendation

When comparing AST solutions, ensure they incorporate developer training as an integral part of their solutions. This will help developers sharpen their software security skills by consuming training within the context of their daily routines and on-demand when it's needed most.

#10 Ensure Onboarding and Deployment Services are Included

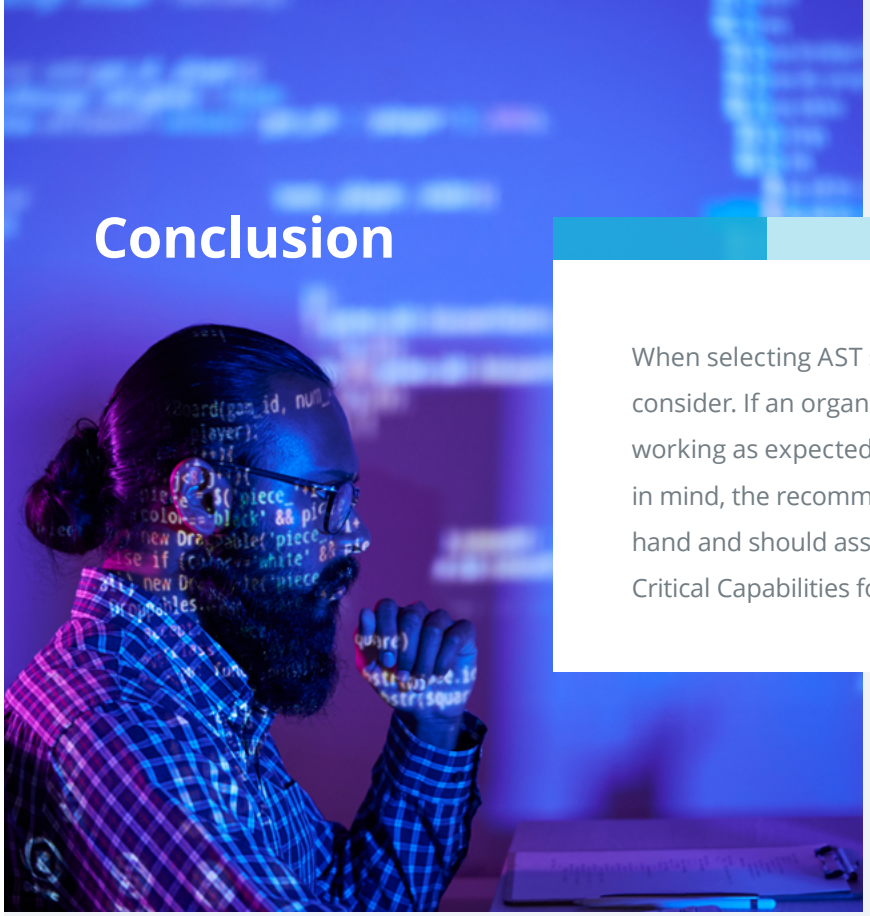
The secret to implementing an efficient, effective application security program is focusing efforts where they will have the greatest impact. In many cases, outsourcing parts of the required activities makes a great deal of sense.

Checkmarx Managed Services let you shift critical, yet costly, aspects of your software security program onto their experts, allowing you to scale effectively and achieve your risk management goals faster. Checkmarx Managed Services include: **Private Hosting** (supporting cloud-based software security initiatives in secure, compliant, private cloud environments) and **AppSec Accelerator** (combining Checkmarx's leading AST solutions with services from dedicated Checkmarx security experts to offload and enhance your Checkmarx software security program.) *Checkmarx Professional Services* focus on addressing critical needs for secure software development and enterprise-class deployment of software security solutions. These include:

- ✓ Onboarding and Deployment Assistance to Drive Adoption
- ✓ Training and Certification
- ✓ Consulting Services
- ✓ Automation and Integration Services
- ✓ Security Program Management
- ✓ Premium Support

Recommendation

When comparing AST solutions, ensure the vendors you're considering also offer a broad range of supplementary and complementary services. Vendor-supplied services are extremely valuable and allow organizations to outsource specific aspects of their AppSec programs where their internal resources may be limited or when they recognize a greater benefit from working with security practitioners to meet their overall objectives.



Conclusion

When selecting AST solutions for your DevOps initiatives, there are many things to consider. If an organization begins moving in one direction, then discovers things are not working as expected, changing directions becomes, at times, unsurmountable. With this in mind, the recommendations in this eBook will help organizations weight the options at hand and should assist them in making well-informed decisions. Awareness of the 10 Critical Capabilities for AST in DevOps is key to getting secure software to market faster.

Software = Security

About Checkmarx

Checkmarx makes software security essential infrastructure, setting a new standard that's powerful enough to address today's and tomorrow's cyber risks. Checkmarx delivers the industry's only comprehensive, unified software security platform that tightly integrates SAST, SCA, IAST and AppSec Awareness to embed security into every stage of the CI/CD pipeline and minimize software exposure. Over 1,400 organizations around the globe trust Checkmarx to accelerate secure software delivery, including more than 40 percent of the Fortune 100 and large government agencies. Learn more at Checkmarx.com