10 Factors to Consider When Embedding AST into Your Pipelines

A Recommendations Guide

Checkmarx

Introduction

Today, software developers and Application Security (AppSec) teams are being pressured to do more with less. The marching orders are simple, “make the software you develop and release as secure as possible—for the lowest viable cost”. This in turn often leads developers and AppSec teams to introduce Application Security Testing (AST) solutions that often fail to deliver the desired results due to a list of unexpected outcomes.

Software security is paramount and application security scans, vulnerability detection, and bug remediation are critical to effectively secure the software your organization develops. This eBook highlights 10 factors to consider when organizations embed AST solutions into their software development pipelines, and it provides straightforward recommendations on what organizations should consider when investigating various AST vendors, solutions, and approaches.

Table of Contents

1. Remediation Guidance is a High Priority

The whole point of AST is not about finding coding errors (bugs) in home-grown applications. Instead, it's all about fixing these errors that could lead to vulnerability exploitation, often resulting in data breaches, account takeovers, and an overabundance of negative consequences. However, many AST solutions are missing key functionalities in the areas of remediation guidance designed to help developers quickly identify where a bug exists. This is basic, yet critical functionality for any AST solution.

Some AST solutions on the market today provide little if any functionality like best fix location that developers find extremely valuable, since it highlights exactly where in the lines of code a bug occurs, and where the best place is to fix that bug within the various lines of code they're working on. Often, best fix location not only highlights the best location to fix a single bug, but also can provide remediation guidance for other bugs detected in latter lines of code. Meaning, if you fix a bug here, you may also fix several other bugs further down the list, when dependencies and interactions are in place between the various code elements.

When bug remediation guidance is missing, developers and AppSec teams are often left on their own to act on the AST scan results and fix the detected bugs, which can be more difficult than expected. As a result, either lengthy delays are invoked to fix the bugs, or the results are ignored due to their complexity and applications are deployed with considerable security issues.

Recommendation

Consider AST solutions that provide your developers valuable remediation guidance like "best fix location".

2. Integration and Automation is of Upmost Importance

In the world of iterative software development approaches like Agile, which are often incorporated directly into DevOps initiatives, workflow integration and automation within the tooling developers and AppSec teams use are imperative. The foundation of modern development methodologies requires these capabilities.

As a result, organizations take complete advantage of automating as much of their processes as possible by integrating various software development and security solutions into their developers' daily workflow. The whole point is of this effort is to reduce the time it takes to develop, secure, deliver, and deploy software. With organizations releasing software weekly, daily, and even hourly, it's easy to understand the importance of integrating and automating the various solutions in use.

Some AST solutions often lack in the ability to integrate into developer workflows out-of-the-box without lots of manual intervention. In some cases, an AST solution may be able to be integrated, but one thing to strongly consider is the time and manpower it will take to perform that task. Instead of using the AST solutions deployed to detect bugs leading to security vulnerabilities, more often, developers and AppSec teams spend vast amounts of time just trying to make the AST solutions work as desired.

Recommendation

Consider AST solutions that provide you with easy-to-implement integration and automation capabilities, partner plugins, and proven approaches.

3. Scan Engine Performance Can Impact Pipelines

Most AST scan processes require software engines that are specifically designed to perform scans that can be quite intensive. Therefore, being able to distribute the scan process across multiple engines is desired since it can directly influence the speed of scan completion. Unfortunately, some AST solutions don't always meet this need since they're often missing a methodology to scale scan processes across multiple engines.

In today's world of multiple developers working on numerous code branches at the same time, having the ability to prioritize scans, reduce average scan queue times, and supporting scan concurrency is highly desired. Scalability is critical to performance, not only in computing, but also with AST.

However, some AST solutions do not support the ability to configure multiple scan engines running on numerous hardware platforms designed to distribute the compute load vertically. In comparison, best-in-class AST solutions support scan engine scaling, improved engine performance, and better DB management that can help reduce scan queue times resulting in shorter overall scan intervals. This ensures that quicker and more-consistent scan times are the result, meeting expectations and requirements much better.

Recommendation

Consider AST solutions that provide you with an approach to easily scale-up and scale-back scan engine conditions that meet your business needs.

4. Incremental Capabilities Improve Scan Velocity

Traditional applications regularly contain millions of lines of code and rarely, if ever, does a single developer have all lines of code in front of them for a given application. As a result, compiling all lines of code into a single application and being forced to launch a full AST scan can have numerous undesirable results. In comparison, modern applications are actually small microservices and often contain no more than 20k-50k lines of code, broken into individual branches based on certain functionalities and requirements. In this case, incremental scan capabilities are a must-have.

Those who work with some AST solutions often report that scans can take way too long to complete since they frequently only support full scans of compiled code, instead of having incremental scan capabilities on uncompiled code. Most understand that when introducing AST into a CI pipeline, there will 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 AST scans taking place to account for the expected delays invoked by the scans.

The real key to increasing scan velocity is to introduce an AST solution that fully supports incremental scanning capabilities on uncompiled code and is designed to only look at the code changes that were submitted since the last full scan. For example, when AST scans are launched incrementally against the branch of code a developer is working on, the results can be returned in a matter of minutes, allowing the developer to use time recovered to work on bug triage and remediation. Leveraging more incremental scans is essential to efficiency and is another important capability to consider.

Recommendation

Consider solutions that deliver incremental scan capabilities to reduce overall scan durations and ensure security does not impact your pipelines.

5. Out-of-the-Box Accuracy is Essential

One of the most significant challenges of introducing any AST solution into a development pipeline is their out-of-the-box inaccuracy. The results of a full AST scan can often return with thousands of potentially detected bugs in the form of critical, high, medium, low, and informational security events. This can cause event overload for developers and AppSec teams who have to trudge through thousands of potential false positives to determine what bugs need fixed and what bugs don't. Moreover, a common "solution" to this challenge is to allocate more humans to improve accuracy, which can be more expensive in the long run.

Another issue related to inaccuracy and high numbers of potential false positives is what happens next. Since a full scan can often returns with thousands of bug-related events, often times developers or AppSec teams do one of two things. They either ignore many of the events and mark the bugs as unexploitable, or they set many of the out-of-the-box queries (rules) to the off position. This usually results in releasing code with potential exploitable vulnerabilities, helps create a false sense of security, and requires organizations to rely more heavily on costly runtime protection mechanisms to protect their vulnerable code.

Some AST solutions come with an extensive list of hundreds of preconfigured queries, grouped together into presets like OWASP Top 10, NIST, CWE/SANS Top 25, and others that often provide good value. Also, some AST solutions have the ability to be updated with new presets and out-of-the-box accuracy improvements. Finally, some AST vendors offer professional services to support users with custom query tuning for unique projects. However, for users of some AST solutions, if a code project includes less-common libraries or custom code elements, tuning queries presets to create a custom analysis can be immensely labor intensive.

Recommendation

Consider vendors who focus on continuous improvements to their out-of-the-box query presets and offer professional services that meet your needs.

6. Integrated Developer Training is a Key Component

AppSec awareness programs that include ongoing and integrated developer security training are essential to reduce repetitive coding errors, resulting in more-secure code. Training designed to keep developers up to date on general AppSec developments, attack trends and methodologies, common coding pitfalls, and secure coding best-practices are vastly needed. At present, most AppSec pros and industry analysts agree that developer training is becoming a base requirement for AST. However, some AST vendors don't offer solutions that provide integrated developer training that's delivered in-band, instead of out-of-band.

Most organizations have observed that lengthy video tutorials, out-of-context training, and monotonous online classes are simply not working. They agree that training needs to be delivered in smaller bits that are relative to the issue at hand, within the tools that developers use. Over longer periods of time, better skills improvement and knowledge retention is the proven result. Unfortunately, some AST vendors leave developer security awareness and training up to the organization to figure out on their own.

Fortunately, best-in-class AST vendors offer just-in-time, gamified secure coding education as part of their solution, enabling organizations to deliver security awareness and training in a more engaging, interactive, and motivating manner that's integrated right within the developers' IDEs. Upon bug detection, developers can jump to a short, relative lesson to the issue at hand, then return to quickly fixing the issue in the code branch they're working on. This approach has tremendous benefits that results in less repetitive coding errors.

Recommendation

Consider vendors that provide continuous developer training that's integrated into the tools your developers use daily.

7. Software Composition Analysis Should be Included

Some organizations may overlook the risk associated with the usage of open source software altogether, since they may believe that open source must be secure because so many people (e.g., the community) are looking at it. On the other hand, some organizations may run their open source code through their static code analysis solution, but this isn't always the best approach. Why? In this scenario, who would fix the overall issue if a problem were found in an open source library that's currently in use?

Best-in-class AST vendors provide static code analysis in addition to integrated Software Composition Analysis (SCA)—used to detect and identify open source components within a code base and provide detailed risk metrics regarding vulnerabilities, potential license conflicts, and outdated libraries. In addition, developers may use open source during development, but those components may not actually be 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. Without this insight from a more-comprehensive SCA solution, organizations will struggle trying to determine what needs fixed and what doesn't in the context of their open source usage.

Some SCA solutions not only can provide detailed risk metrics, but they can also identify both direct and transitive dependencies and provide 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. AST vendors that include an integrated static code analysis and SCA solution provide the best method to thoroughly secure software overall.

Recommendation

Consider vendors who provide integrated static code analysis and SCA to thoroughly address risks throughout your entire code base

8. Onboarding is More Important than Most People Think

When organizations choose any AST solution, services should be available to help them get up and running quickly, and the "you're are on our own" approach can rapidly become overwhelming. Unfortunately for most organizations, this is where they often experience major setbacks, frequently leading to complete breakdown of their software security initiatives.

It's well known that AST solutions are just as complex as the software they're designed to help secure, and if any solution is overly easy, it's usually lacking in efficacy that further reduces the overall AST value. To worsen the predicament even further, some vendors offer little help with onboarding their solutions, resulting in considerable amounts of unplanned expenses in the form of inhouse manhours used to get solutions up and running.

Any organization's software security goals must be fairly modest at first. For example, they may simply want to address the most critical vulnerabilities in their code base and get more proficient over time. Or they have a requirement to address the OWASP Top 10 or maybe the SANS Top 25 and need help getting there. Outsourcing some of that effort to a best-in-class AST vendor that offers onboarding services is a good option for organizations to help them get started and move forward rapidly.

Recommendation

Consider including vendor-provided onboarding services into your AppSec initiatives to help you get up and running quickly.

9. You Don’t Always Get What You Pay For

Organization that try to implement sub-par AST solutions usually do it for these reasons: budget constraints and/or lack of funds, a checkbox for upper management, to satisfy some board initiative, or as a checklist item for some form of regulatory compliance. And sometimes smaller organizations take a low-cost approach simply because decision makers don’t really know what other AST options are available and are not privy to benefits vs. costs analyses.

The point they often miss is that best-in-class AST solutions actually reduce development time, improve security, and enhance delivery and deployment frequency. This tremendously offsets the hidden costs of sub-par solutions. In addition, for any AppSec program to be successful, it requires developer adoption of AST. This is a fundamental requirement. If the AST solutions you chose end up disturbing developers’ workflows, causing them to miss their deadlines, they will simply resist using them overall. In turn, this will give developers, AppSec teams, and management a bad impression of your AppSec approach overall.

Recommendation

Consider best-in-class vendors and solutions that actually reduce development time, resulting in faster, more-secure releases.

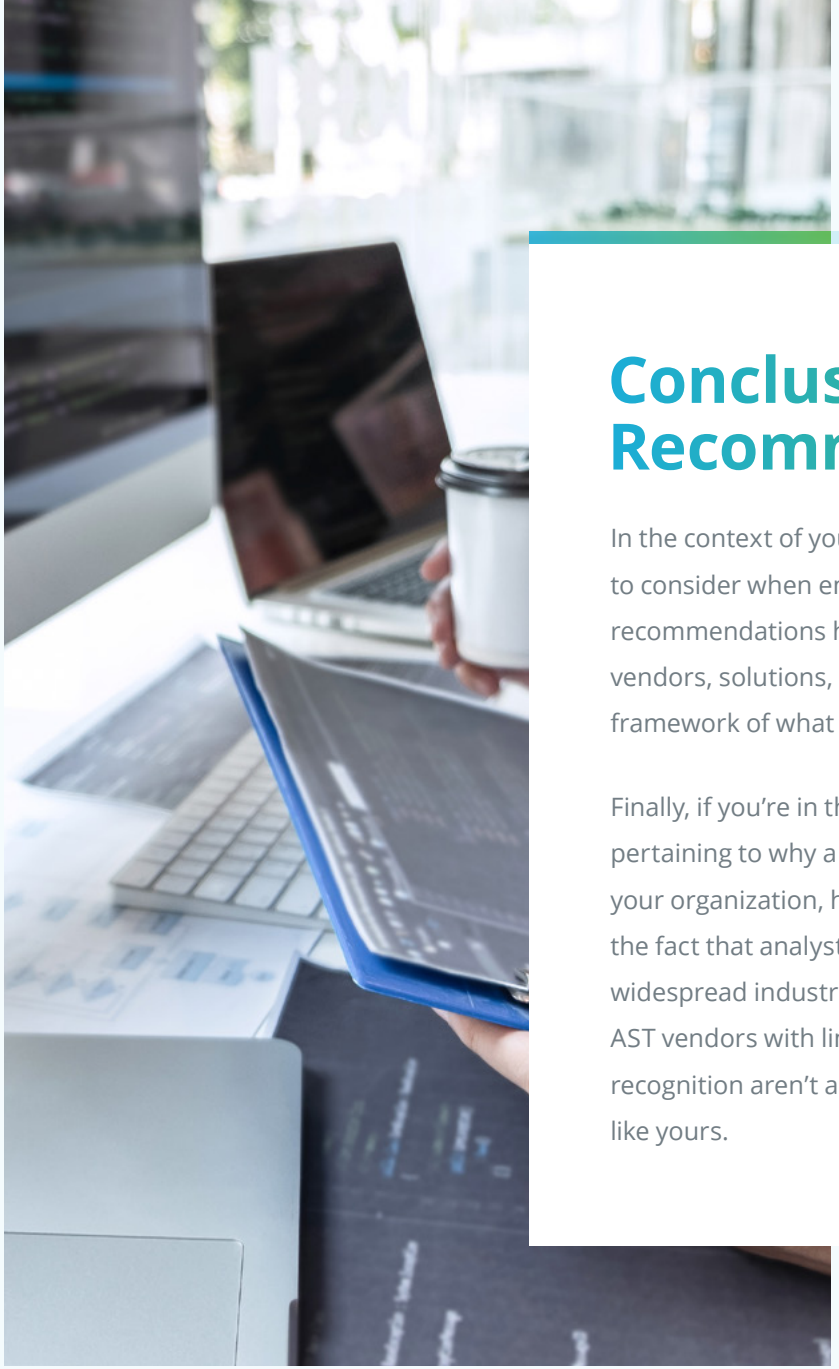
10. You May Struggle When Going It Alone

Regardless of the motivation for AST solutions, going it alone is not always the best recommendation and the right, specialized AST vendor can help organizations build upon what they're already doing best—building software to solve a problem and/or generate revenue. What organizations want are solutions that help build better software that's more secure, and this is where managed services can help.

Well-defined services help organizations quickly get started, especially when vendor teams have vast amounts of AppSec experience. These services include solution onboarding, query preset and scan engine tuning, solutions training, AppSec consulting, AppSec risk analysis, fully managed alternatives, and many other applicable options. Most often these services are rather affordable and can easily offset the consumption of more expensive inhouse manhours needed to get AST solutions working proficiently. In other words, there are valuable AST services available that won't break the bank.

Recommendation

Consider working with an AST vendor who has vast amounts of AppSec program experience and expertise that also includes services you may need.



Conclusion and Final Recommendation

In the context of your AppSec initiatives, hopefully this eBook brought light to the 10 factors to consider when embedding AST into your development pipelines. In addition, the short recommendations herein were designed to be used as criteria when evaluating various vendors, solutions, and approaches. The 10 factors were meant to help provide you with a framework of what to contemplate during the decision-making process.

Finally, if you're in the position of trying to explain your reasoning to decision makers pertaining to why a best-in-class AST vendor, solution, and approach may be a better fit for your organization, highlight the 10 factors to consider as shown in this eBook. Also highlight the fact that analyst firms, customer validations (via referenceable testimonials), and widespread industry recognition are also key indicators that you're making the right choice. AST vendors with limited or no support options, few, if any testimonials, and no industry recognition aren't always the right choice for today's growing, software-driven organizations like yours.

Addendum

About Checkmarx Managed Services

Checkmarx Managed Services let you shift critical, yet costly aspects of your software security program onto our experts, allowing you to scale effectively and achieve your AppSec goals faster. Checkmarx managed services include:

Private Hosting: Supporting our customers' cloud-based AST deployments in secure, compliant, AWS private cloud environments, fully managed by Checkmarx solution experts.

AppSec Accelerator: Combining CxSAST, CxSCA, CxIAST, and/or CxCodebashing (Secure Coding Education) solutions, fully managed by our developer and AppSec experts to offload/enhance your organization's AppSec program.

Checkmarx Professional Services puts a laser focus on addressing your most critical needs to quickly and measurably improve your AppSec program, plus enterprise-class deployment of our software security solutions. These services can include:

- Onboarding and Deployment Assistance to Drive Developer Adoption
- Solution Management and Operations Training
- Checkmarx Solution Certifications for Your Employees
- Consulting Services for Your AppSec Program Development
- Automation and Integration Services into Your Developer Tooling
- Security Program Management and Ongoing Support
- Premium Support and Technical Account Managers