Static Application Security Testing: A SAST Tools Buyer's Guide

Table of Contents

Overview 3

Who Needs SAST? 3

Why is Choosing a SAST Tool an Important Decision? 4

What Can SAST Do? 4

Benefits of SAST 5

SAST Tool Requirements 5

Critical Questions for SAST Providers 7

Conclusion 9

Overview

SAST, or static application security testing, is a tool that analyzes code or binaries in order to detect security flaws. A good SAST tool integrates with development workflows, allowing for automated testing and quick feedback to developers on issues that need to be fixed. It helps make software security a more integral part of software development, thus putting better, more trustworthy software into clients' hands.

Who Needs SAST?

The truth is: all software code includes bugs. But there are different kinds. Some break features so the software doesn't work the way it should. And some introduce attack vectors that a malicious user can use to get into your systems and anything the application interacts with, like user data and compute resources. Because of this, any organization that develops software should use a SAST tool to help find these security issues.

Even if you have other testing tools, like dynamic application security testing (DAST), having SAST in your arsenal is immensely beneficial. DAST tools are good at detecting issues like runtime vulnerabilities and insecure environments, since those require fully functioning programs to test for. However, for many other common issues having to do with data input, coding best practices, or even (thanks to **next-generation SAST features)** data flow, SAST can help you identify problems while the software is being written, even before there is a full, compiled application or feature.

Why is Choosing a SAST Tool an Important Decision?

When you develop software, the SAST tool you choose will be an integral part of your ability to assess the security of what you make. Your reputation is at stake, as is the trust from your clients and customers. Furthermore, it is a significant investment in both time and money; you want to get the most for your investment. You also want to make sure that the SAST tool you choose meets your needs now, and will be able to grow and change with you as your business moves into the future.

Before choosing a SAST tool for your company, it is critical to know what you plan to get out of SAST, know how SAST will fit in with your development processes and your future plans, and ask the right questions of any SAST vendor you are considering.

What Can SAST Do?

The core purpose of SAST is to identify security issues in software. Compared to DAST, which requires fully compiled and operational code, SAST can help you find security issues from source code or binaries. It can be integrated at multiple points of the development cycle: at each commit, when builds are completed, and when builds are verified. Modern SAST tools can identify more issues than ever before, with the ability to not only look at individual lines or functions, but also use data structures to map out data flow through the application. SAST helps integrate software security throughout the cycle, and puts developers in the position to fix issues sooner, or in the moment, instead of waiting for the issues to become technical debt.

Of course, SAST is not the only tool you need for software security. It is an important layer, but as with most security challenges, defense in depth is your friend. The good news is that modern, full-featured SAST can help you go deeper into your software. It can integrate with other software security processes to build a holistic view of your application security, including analysis of both first-party and third-party code. This is important because, in addition to the code your developers write, modern applications depend heavily on third-party, open source libraries, which allow them to quickly implement common functions instead of building them from scratch.

Benefits of SAST

SAST is at the core of shifting your software security left in the development cycle. Unlike with dynamic tools, evaluating software security with SAST does not require a full build, only source code or a binary. Code can be evaluated with SAST throughout the development process, **as early as commit-time** with an incremental SAST implementation. Many even include integrations with the development environment, which allows for analysis before any code leaves a developer's machine. This means SAST can help your developers identify and correct software security issues as early as possible during the development cycle, instead of only when an entire tool or feature has been created. It helps you save time and money on secure development, and ship more secure code more quickly

SAST helps you find out how well your software is equipped to resist many threats, including both wellknown ones and emerging issues. Like any good vulnerability detection tool, full-featured and actively maintained SAST should help your software resist a broad spectrum of attacks. It is also able to **integrate** with vulnerability databases and threat intelligence feeds. This means that SAST goes beyond old, familiar software vulnerabilities. It can flag software that is vulnerable to current attacks, keeping both your software and your developers more ready to face an ever-changing threat landscape

SAST Tool Requirements

Before selecting a SAST tool, you can ensure as smooth an experience as possible by identifying your requirements at the beginning of the selection process. This will help you narrow down your list of vendors and formulate the right questions as you gather information. These major categories of requirements that businesses consider when selecting a SAST tool:

DETECTION ACCURACY AND PRECISION

Any security testing will generate additional work for developers when it comes time to fix vulnerabilities.

Because of this, you want to make sure that the issues your tool finds are real, so developers' time isn't wasted. When considering the vulnerability detection capabilities of a SAST tool, you will want to minimize false negatives and false positives while maximizing true positives. A **false negative** refers to when SAST fails to detect a problem when there is actually a vulnerability in code. Though a lighter-weight scanner may give developers less to correct, it also leaves open the probability that more vulnerabilities will make their way into software releases. This, in turn, can lead to damaging and expensive breaches for those who are using your software.

A false positive, on the other hand, is a finding that does not actually indicate a vulnerability. High false positive rates can reduce trust in the SAST solution and lead development teams to disregard results. False positives also require investigation on the part of developers, taking their time away from innovating and doing the work they're passionate about. On the other hand, false positives can happen, and they are generally less damaging than false negatives. Keep in mind that you can tune SAST to reduce the false positive rate. But, reducing false positives should not come at the expense of actually finding vulnerabilities in code: in other words, detecting true positives and minimizing false negatives.

INTEGRATION

The best security tool is one that your team will actually use. In that spirit, SAST is most useful when it is a seamless part of the development process. The right SAST is going to integrate with the infrastructure that you already use to manage your processes, including CI/CD tools, ticketing tools, version control, and code repositories. Depending on your level of security maturity and regulatory oversight, you will likely also need SAST results to be imported into other tools such as ERP, SIEM, or SOAR for auditing purposes. Ensuring that SAST integrates well with the tools you use before choosing it can save significant time and money along the line.

FUTURE BUSINESS NEEDS

Your business does not remain static, and neither should your SAST tool. In addition to nuts-and-bolts software testing features, a SAST tool should be able to evolve with your business. Make sure, while considering your SAST options, you evaluate them not only in the light of what your software development tools, technologies, and processes look like now, but what they will look like in the future. If you are planning changes of scale, language, or security frameworks, make sure that the SAST tool you are considering will work with the ones that have plans, or have trustworthy plans to be compatible. Then, you can evaluate the tools you are considering — and the companies behind those tools — in light of how they can satisfy those considerations.

Critical Questions for SAST Providers

As you assess your SAST options, be ready to ask important questions to the providers you are considering. Before committing time and money to SAST for your business, make sure that you know these things.

PRICING MODELS

As with any development or security tool, knowing what the real cost of a tool will help you both budget and make the business case. Pricing models vary by provider. Common ones include pricing per scan, per user, per server, per core, or per line of code scanned.

Before signing a contract, have an idea of what you expect to need based on these common pricing models. Ask any provider you are considering what their pricing models are. Then, be ready to analyze it in the context of your own business, both as it exists now and as it may be based on plans to grow or change in upcoming years.

SAST DELIVERY

Delivery of SAST tools varies by vendor and offering. Given the pros and cons of each method, be ready to ask the providers you are considering about what model they use.

Some tools are designed to run on premises or on physical hardware. These can raise questions of costs for acquiring or upgrading hardware in order to meet your software security scanning needs. So, when considering SAST designed to run on a physical server, make sure to ask about the processing needs, storage requirements, and typical hardware installations that are required to run it at a scale that fits your business.

Some tools are designed to run on cloud instances. This raises questions of the cost of cloud instance size or processing power. Analogous to SAST designed to run on physical hardware, be sure to ask about the scale of cloud instances or cloud service processing power that is required to serve your business, both as it exists now and as you expect to grow.

Other SAST tools run on Software-as-a-Service (SaaS) platforms. SAST delivered via SaaS overcomes the questions of cloud or physical capacity cost. However, in addition to the pricing model, it requires asking about the security practices of the provider, especially if your code has to be uploaded to the SAST provider’s platform.

VULNERABILITY AND THREAT DATABASES

Before choosing SAST, make sure you know what the tool can detect. A good SAST tool must be able to grow with your software and the threat landscape by defending against not only the classic categories of vulnerabilities, but also the leading edge of attacks against software. Make sure to ask what vulnerability databases SAST integrates with, and how frequently detection is updated based on changes in those databases and the threat landscape.

BENCHMARKING

SAST vendors should be able to tell you how their scanners measure up against industry-respected measures such as the **OWASP benchmark.** They are designed to test how well SAST can detect common, real vulnerabilities that affect users' software security posture. The information a benchmark can reveal includes both **false negative** and false positive rates. This helps you determine a tool's ability to assess the security of your software while also making the most of your developers' time.

PARALLEL SCANNING CAPABILITIES

If your company is developing software at scale, your SAST needs to be able to keep up. If different teams are working on different applications, or different features, the tool needs to be available whenever developers are ready to commit code. Otherwise, SAST is going to slow down the software development process.

Before choosing a SAST tool, ask the provider whether their tool is equipped to run parallel, or concurrent, scans. Find out how many can be run at once, and what effect that may have on the speed of the scans. After all, SAST should help you test software security at the scale you need, and the scale you may need in the future, depending on your goals.

LANGUAGE SUPPORT

Most SAST providers will provide a list of languages that their platform is designed to cover. This is important to know, but not enough to paint **the entire picture.**

All language coverage may not be created equal. A SAST tool may have better developed detection capabilities for some languages than others, leaving you with more false negatives and false positives for languages where their capabilities are less refined. Ask the vendor not only about the languages they cover, but also how their testing in the languages your business uses stands up to benchmarking. It also helps to ask about their future language coverage roadmap, whether they add languages upon request, and what their timeline is for adding languages upon request. This can help you assess their flexibility, responsiveness, and ability to suit your company's needs as you grow and change.

INTEGRATION AND RESULTS FORMATS

Before choosing a SAST tool, you should ask what integrations are available, and ensure that it works with the tools you currently use and any that may be on the horizon to adopt. SAST is only as useful as what your developers can see and address. Developers are more likely to see and address things that show up in the tools they are already accustomed to using to manage the development process.

As with language support, it is also worth asking for information about future planned integrations and handling of custom requests for integrations. This can give you a good idea about whether a SAST vendor is ready to grow and change with your business

Conclusion

Choosing a SAST tool is an important decision. It is one of the core elements of secure software development, and it is a tool that you will depend on for years to come. You will save time, money, and headaches — and also put yourself in the best position to develop secure software — by assessing your needs fully and choosing a SAST tool that will fit them for the long term.

If you are serious about a SAST tool that can help you address security issues sooner in your software development process, Qwiet AI may be right for you. Qwiet AI is SAST delivered via a SaaS platform. It offers a broad range of language compatibility, and its Code Property Graph (CPG) technology gives it the ability to map routes across modern, modular applications.

Visit Qwiet.ai to book a demo, or to create a free account and experience our platform for yourself.