Lionfish
Tech Advisors

FOURCASTERS
BECAUSE WHAT HAPPENS NEXT MATTERS

Application Security Testing

Market Overview

The Application Security Testing (AST) market is mature, but the boundaries of the market are changing. Adjacent security platforms, increased software supply-chain exposure, and the rapid adoption of artificial intelligence in software development are changing where and how application security testing occurs. These changes may alter how AST is purchased and delivered, but they do not eliminate the underlying need for independent testing, flaw lifecycle management, and validation.

> ***This distinction between providing individual testing capabilities and providing enough of the components needed to support an application security program is important throughout this research.***

THE PYRAMID for Application Security Testing includes vendors that offer enough of the components needed to build and maintain an application security testing program. For the purposes of this research, these include Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA), reporting and trending, secure code training, remediation assistance, and pentesting services. Vendors offering only one or two testing types, or that substantially lack the completeness or scale needed to support an AST program, are excluded. This research intentionally narrows the included vendor aperture to provide value for organizations with the objective of building a program in mind.

This distinction between providing individual testing capabilities and providing enough of the components needed to support an application security program is important throughout this research.

Forces adjacent to the Application Security Testing market over the last few years have led to speculation that the segment would become obsolete. Specifically, the rise of Cloud Native Application Platform Protection (CNAPP) was supposed to obviate the need for AST if you believed the marketing architecture where AST held only a small and seemingly insignificant spot in a much larger, end-to-end, holistic solution. Then, of course, artificial intelligence flaw-finding and fixing should also eliminate AST. None of these things have occurred. In fact, the rumors of the demise of AST as a business function have been seriously incorrect.

Artificial Intelligence Creates the Need for Continuous Testing

Within the application development and ongoing maintenance ecosystem AI is having a massive impact. AI is helping programmers and organizations produce more code at higher velocity, but it still cannot reliably produce flaw-free code. Because of the volume, the need for AST scanning has only become more pressing.

The larger change may be that AI moves AST from a periodic testing activity toward an always-on validation layer. AI-generated code can be created and changed continuously, and AI-driven remediation creates additional code changes that also then need to be validated.

To form an effective program, AST scan types are layered together to form an overall blanket of coverage, combined with human pentesters and application security staff helping to remediate flaws before (or after) they go to production. To be sure, if AI finds hundreds of flaws that is all well and good. Finding is not fixing, and fixing is not validation. In short, simply trusting that some future AI solution can write flaw-free code or find all the flaws fails the need for redundancy that forms the foundation of an AST program.

The same AI system or model family used to create cannot be used to remediate the same code and should not automatically be considered an independent security control. Even highly capable AI remediation produces a code change, and that change requires validation before the flaw can reliably be considered resolved.

AST vendors have introduced the use of AI to assist in finding flaws as well as fixing the code involved in some of the more common CVEs. This can materially assist in reducing security debt that naturally accumulates over time as applications age. Reducing security debt is a non-trivial improvement in an AST program and is a very desirable addition for any program as part of the vendor selection process.

If these capabilities continue to improve, AI may substantially reduce the human effort required to triage and remediate findings while simultaneously increasing the frequency with which application security testing is performed. This would change the operation of AST considerably without eliminating the AST function itself.

CNAPP and ASPM Expand the Competitive Perimeter

Cloud Native Application Platform Protection (CNAPP) vendors may provide a seemingly end-to-end solution; however, ingesting results from AST vendors or even having a few scanning types within a solution is not an application security program and this distinction is important. In a successful application security program, flaws found must be assigned, owned, remediated, certified, or accepted and recertified. This business process is the nexus and often the key point of cooperation between development and security organizations and not merely a technology that is deployed. People are still needed to understand and own the risk if fixing a flaw is deemed unnecessary. This decision could be reached, for example, if exploitation of an identified flaw is unlikely or a scanner finding is a false positive. Similarly, secure coding is a program, not a feature. A successful AST program includes secure code training, reporting, trending, and vendor- or community-provided remediation assistance. In short, demonstrating how the organization is doing and that they are improving and making progress.

The promise of ASPM, however, depends heavily on accurately correlating security findings. This is substantially more difficult than simply ingesting and normalizing results from multiple tools. An application may contain the same flaw pattern in multiple locations, while normal code changes can move or alter an existing flaw without changing the underlying security issue. Results from different testing products may also describe the same underlying defect differently. Accurately determining whether a finding is new, previously identified, duplicated by another testing source, remediated, or simply changed in location is therefore critical plumbing for application security posture management.

This becomes particularly important when workflow and governance decisions are attached to a flaw. Ownership, remediation history, false-positive determinations, accepted risk, certification, and recertification must remain associated with the correct finding over time.

Poor flaw matching can either recreate previously dispositioned findings as new noise or, more seriously, associate an existing risk decision with a finding that is actually new. This is one reason delivering the full promise of ASPM is more difficult than the architectural concept suggests.

For CNAPP vendors entering this area, application and runtime context can provide considerable value in determining which flaws represent meaningful exposure. That does not, by itself, replace the underlying testing engines or the application security program. An ASPM or CNAPP control layer may increasingly become the place where risk is viewed and prioritized while still depending upon AST technologies underneath it for detection, flaw identity, and validation.

While some day CNAPP vendors may be able to provide some of the necessary components that make up the AST segment they are not there yet and for this reason CNAPP has been excluded from this research. ASPM capabilities alone are similarly insufficient for inclusion unless the vendor also provides enough of the underlying testing and program capabilities required by the research definition.

> ***Poor flaw matching can either recreate previously dispositioned findings as new noise or, more seriously, associate an existing risk decision with a finding that is actually new.***

The AST Perimeter Is Beginning to Expand

The boundaries of Application Security Testing are beginning to expand, although this is not happening equally across vendors or customers. API testing, software supply-chain security, secrets detection, infrastructure-as-code testing, runtime context, and automated remediation are increasingly included in the offerings in the traditional AST market. Many AST vendors are already pushing into adjacent areas as they posture for competitive advantage. Veracode, Snyk, Checkmarx, and Black Duck, for example, provide ASPM capabilities through acquisition, internal development, or both. The direction is toward a broader and more continuous function than the traditional combination of SAST, DAST, and SCA.

Over time, this may result in a more continuous, always-on model where testing, prioritization, remediation, and validation happen throughout the development lifecycle. Whether that is a human or an agent in a workflow is less relevant than whether it gets done. This is not where every customer or the vendors providing these future workflows is today. However, higher development velocity, AI-generated code, and increasingly automated remediation are likely to make a less integrated and continuous testing process less practical. The need for testing does not go away; its frequency and operation change.

AST Is Evolving, Not Disappearing

Application Security Testing is a mature but evolving market, and it is reasonable to expect continued consolidation and movement into adjacent security categories. It is also possible that over time some of the capabilities associated with AST will be purchased as part of larger development or security platforms rather than as standalone products. This is different from saying that the AST function itself is becoming obsolete.

AI is increasing the amount and velocity of code being produced while also beginning to automate remediation. CNAPP and ASPM platforms can add valuable application, cloud, and runtime context. None of these developments eliminates the need to independently identify flaws, understand which flaws matter, assign ownership, remediate or accept risk, validate the result, and demonstrate that the application security program is improving over time. These are business functions and not simply scanner features.

The more likely future is not the disappearance of AST but a change in how the function is delivered. Testing may become increasingly continuous and automated, while the perimeter of what is considered application security continues to expand. Which vendors ultimately control that workflow is still very much in play. The underlying requirement to independently test and manage application risk is not.

Application Security Testing

THE PYRAMID

THEPYRAMID™

Powered by The Aquarium™

APPLICATION SECURITY TESTING

Buyer's Guide 2026