Interactive Application Security Testing 101

How to Evaluate and Implement an IAST Solution

Table of contents

The who, what, and why about the next big thing in AppSec

Many are hailing interactive application security testing (IAST) as the next step in the evolution of application testing, and for good reason. As more organizations adopt DevSecOps, they'll need to consider how traditional application security testing tools fit into this new paradigm. A 2019 Gartner report on DevSecOps recommends IAST as an alternative tool, stating, "IAST incorporates attributes of both SAST and DAST, leveraging instrumentation of the application during testing. … This combination enables IAST approaches to provide a better balance of efficacy—the reduced false positives of DAST with the precise line of code and code coverage visibility of SAST."[1]

IAST provides some distinct advantages over traditional application security testing methodologies. In this eBook we introduce this exciting new technology and explain why it's set to disrupt the application security testing world. Continue reading to see why no software security toolkit is complete without an IAST solution.

What is IAST?

Interactive application security testing is a software security testing technique that analyzes the behavior of web-based applications as they run. IAST solutions typically work by deploying agents in a running application. These agents continuously analyze the application's interactions (usually initiated by automated tests) to identify security vulnerabilities. IAST sees every line of code as it is executed and the stack trace, memory values, and actual dataflow of an application as it responds to each HTTP(S) request. Some IAST solutions can not only actively monitor security vulnerabilities (e.g., SQL injection) but also verify them and show that they are real and exploitable. Then they produce a vulnerability report with line-specific remediation advice, empowering developers to fix actual prioritized vulnerabilities immediately.

One of the features that define IAST is where it is normally implemented in the SDLC: IAST typically runs in the integrated test and QA stage. By pushing security testing left from production (where DAST usually takes place), teams can catch runtime vulnerabilities earlier, thereby reducing remediation costs, eliminating delays, and reducing the risk of breached applications.

The best IAST tools provide integration with software composition analysis (SCA) tools, which can scan binary files for third-party and open source components and report known vulnerabilities associated with those components, as well as their associated licenses and other valuable information.

Why should I care about IAST?

Web applications run by large organizations are ideal attack vectors for hackers wishing to access sensitive personal data, intellectual property, and more. According to the 2019 Verizon Data Breach Investigations Report, web application attacks remain one of the top attack vectors across multiple industries.[2] These breaches often cause significant financial damages and long-term damage to a business's reputation. Here are just a few recent examples:

- 880,000 payment cards were exposed on travel booking site Orbitz.[3]
- Three months' worth of payment data was stolen from Rail Europe's American website.[4]
- Up to 40,000 customers had their payment information taken from Ticketmaster's U.K. website.[5]
- And of course, the infamous Equifax breach resulted in more than 145 million people's personal data being exposed.[6]

Though Equifax happened in summer 2017—what seems like ages ago in the world of cyber security—it continues to resonate, not only because of its wide-ranging impact but also because it would have been so easy to prevent.

Traditional application security testing methods provide a layer of defense by helping you find and fix potential vulnerabilities, but they differ in how they scan and test applications. Some are better than others at finding these threats depending on a host of conditions, including the testing environment, the stage in the software development life cycle (SDLC), and general use cases.

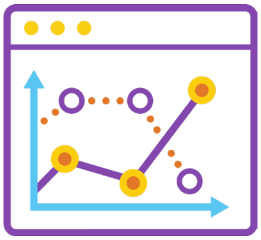
These differences create a dilemma for organizations: If they do not select the right tools, they might unwittingly expose their web apps to an unforeseen attack. Fortunately, IAST solutions lessen these headaches for organizations by shifting testing left, so problems are caught earlier in the development cycle, reducing remediation costs and eliminating delays—all without disrupting normal workflows.

Security teams need tools that can give them a continuously updated view of the risk posture of their web apps and compliance with security standards *before* those apps are deployed to production.

How is IAST different from other AppSec tools?

To keep pace with the demand for rapid development of web applications, organizations need accurate and automated security testing tools that can easily scale and produce actionable results. Static analysis, the most prevalent application security testing solution today, provides comprehensive analysis of static source code, but can't identify runtime vulnerabilities found via dynamic testing. While static analysis tools typically detect a high number of vulnerabilities and produce accurate true positives, they also generate a high volume of false positives. This is because testing occurs early in the coding and development phase as opposed to detection at the application binary runtime level.

Alternatively, dynamic and manual testing find vulnerabilities at application runtime and help reduce the volume of false positives. However, they do not provide much detail on the vulnerabilities they detect, or remediation advice. These vulnerabilities require additional review and validation cycles by security experts, increasing the workload of already strained security resources and development teams.



Dynamic application security testing

DAST tools test running applications from the outside in by attacking them externally. Coverage is limited because DAST solutions are essentially blind as to what is happening inside an application. Challenges include moderate false-positive rates, an increased number of testing cycles, and increased testing duration. Finally, DAST results offer no code guidance as to where software vulnerabilities are located, making it difficult for developers to easily fix identified vulnerabilities. DAST tools can't effectively achieve the fast turnaround times required in CI/CD workflows, unlike IAST, which produces real-time results and detailed insights for timely remediation.

Static application security testing

Alternatively, SAST solutions are great at identifying security weaknesses and providing code-level guidance as to where and how to fix vulnerabilities in source code. And they provide integrations for developer IDEs, issue trackers, and build tools to support CI/CD workflows. But SAST is blind to how all the pieces of an application work together and operate at runtime, so it can't detect vulnerabilities in running applications that hackers may be able to exploit. In addition, SAST reports can be overwhelming, identifying many potential vulnerabilities.

Interactive application security testing

IAST fills the gap between traditional static and dynamic testing and is a great complement for teams adopting DevOps and continuous integration and continuous delivery (CI/CD) practices. IAST solutions allow users to find and fix security vulnerabilities using real-time data and work in running applications. They quickly identify a broader range of runtime vulnerabilities with greater accuracy than DAST and SAST solutions, and they do so down to the line of code that should be fixed. Some IAST solutions include SCA and e-learning, which enable teams to learn about and fix vulnerabilities in third-party and open source components easily. Because IAST provides real-time results in mere seconds, it is the only type of dynamic runtime testing that can support DevSecOps and quick-turnaround CI/CD processes. It can also be easily integrated into the existing development and testing cycle, allowing security and development teams to expend their time and energy on true positives that matter.

How can I benefit from IAST?

1. Actionable findings for development teams

In a report by Forrester, IAST was shown to reduce the time it took to remediate security vulnerabilities by 65%, compared to penetration testing.[7] This is because IAST empowers developers to find and fix vulnerabilities as a part of their development process. Application security experts can remove themselves from the critical path of software development and spend more time on strategic security initiatives.

2. Comprehensive vulnerability and security risk reporting earlier in the SDLC

IAST enables developers to fix security vulnerabilities as they code, reducing reliance on external security testers for pen testing. This means you can find and fix runtime vulnerabilities in web apps before deploying them to production. Shifting left and doing security testing earlier in the integrated build and testing stages enables substantial cost and resource savings for organizations, while also reducing security risk.

3. Low false-positive rates

IAST solutions are automatic and accurate; they won't return long lists of potential vulnerabilities that require lengthy, tedious manual review to resolve and eliminate false positives. So organizations can focus DAST and pen testing budgets on more difficult corner-case vulnerabilities that require more intensive manual human testing to identify and verify.

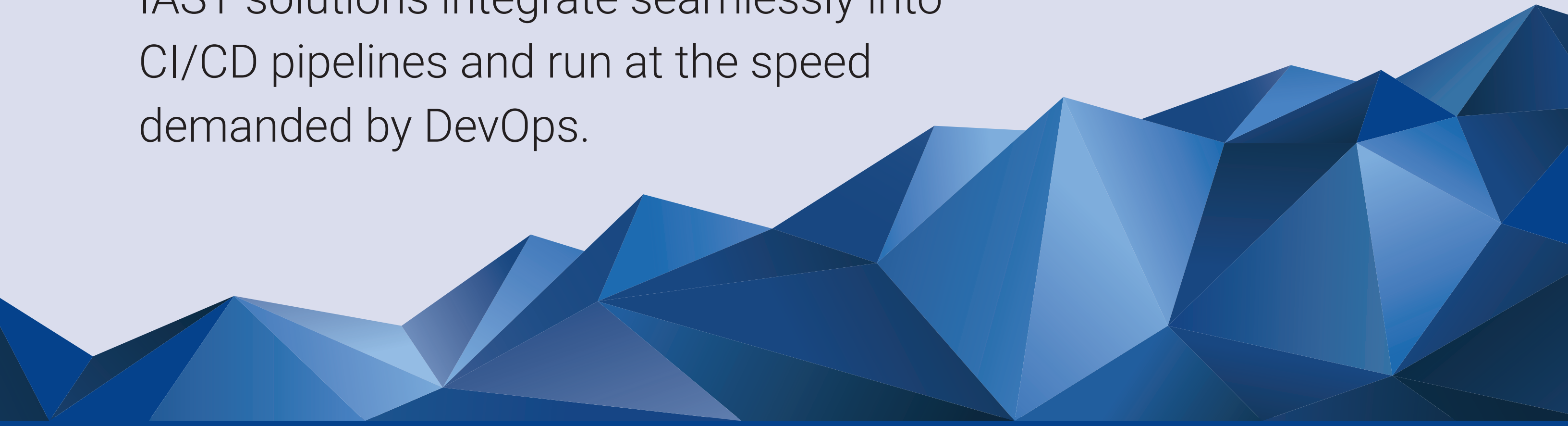
4. Seamless integration into automated development and testing environments

If development teams are to adopt security testing as part of their normal workflows, an application security solution must be able to plug into and integrate with agile and CI/CD development tools. It also must be easy to deploy, update, and scale to support large enterprise requirements. IAST solutions integrate seamlessly into CI/CD pipelines and run at the speed demanded by DevOps.

Both security and development teams can benefit from integrating IAST into the SDLC, especially an IAST tool that includes SCA and e-learning. Security teams need application security tools that can comprehensively find vulnerabilities and give them a continuously updated view of the risk posture of their organizations' web apps and compliance with security standards. And they need this information before web apps are deployed to production, where they're at risk of security attacks that may lead to costly data breaches.

Development teams, by contrast, need quick feedback on what vulnerabilities to fix, how to fix them, and where to find them in their source code or component libraries. And developers need this feedback early in the SDLC, when they're most familiar with their code and when vulnerabilities are least costly to fix.

IAST solutions integrate seamlessly into CI/CD pipelines and run at the speed demanded by DevOps.



What should I look for in an IAST solution?

IAST has many distinct advantages over traditional solutions, which is why it's poised to be the next big thing in the security industry. No matter what solution you choose, we recommend you consider the following:

Must-have	Why it's important
1. Updated security dashboards for standards compliance: PCI DSS, GDPR, OWASP Top 10, SANS/CWE	You need insight into security risks, trends, and coverage, as well as security compliance for running web apps (including custom code and open source components).
2. Fast, accurate, and comprehensive results out of the box, with low false-positive rates	You need to spend less time finding and remediating false positives. But you can't waste time configuring and tuning your tools to reduce them.
3. Automated identification and verification of vulnerabilities	You want to free up your teams to find and fix more complex vulnerabilities. So you need a tool that verifies each vulnerability and doesn't inundate you with false positives.
4. Sensitive-data tracking (e.g., PII and company IP)	You need to achieve compliance with key industry security standards (e.g., PCI DSS and GDPR) by setting parameters to automatically track sensitive data in applications.
5. Ease of deployment in DevOps and agile workflows	Your web app development and DevOps teams rely on agile development and automation. So they need AppSec tools that seamlessly integrate with standard build, test, and collaboration tools and "just work."
6. Enterprise-grade SCA binary analysis integration	You need visibility into security vulnerabilities and license types and versions in open source and third-party components, libraries, and frameworks.
7. Real-time insights, detailed remediation guidance with contextual learning	Your developers need detailed information about vulnerabilities and where they are located in their code, as well as contextual guidance on how to remediate them.
8. Ability to scale and support modern development and deployment	You need an IAST solution that can easily scale, bind together multiple microservices from a single app for assessment, and support container deployment.

Why Seeker?

Seeker is an enterprise-scale IAST solution that fits seamlessly into CI/CD development workflows. Easy to use and deploy, Seeker can quickly process hundreds of thousands of HTTP(S) requests with extreme accuracy. Unlike other IAST solutions, Seeker uses active verification to automatically validate whether each identified vulnerability is exploitable, reducing false positives to near zero and providing tremendous time and cost savings. Seeker's innovative sensitive-data tracking feature, the first in the industry, provides the utmost visibility into where your most critical information is stored with weak or no encryption, helping you ensure compliance with PCI DSS, GDPR, and other security standards and regulations.

References

1. Gartner, Neil MacDonald and Dale Gardner, 12 Things to Get Right for Successful DevSecOps, Dec. 19, 2019.
2. Verizon, 2019 Data Breach Investigations Report, 2019.
3. Dani Deahl, Orbitz Says a Possible Data Breach Has Affected 880,000 Credit Cards, The Verge, March 20, 2018.
4. Zack Whittaker, Rail Europe Had a Three-Month Long Credit Card Breach, ZDNet, May 14, 2018.
5. Taylor Armerding, GDPR Raises the Stakes on Data Breaches, Synopsys Software Integrity Blog, July 12, 2018.
6. Lily Hay Newman, Equifax's Security Overhaul, a Year After Its Epic Breach, WIRED, July 25, 2018.
7. Amy DeMartine, Construct a Business Case for Interactive Application Security Testing, Forrester, Nov. 2017.

Shift left with Seeker

For more information on how Seeker can help you shift left and optimize your application security testing strategy, visit synopsys.com/iast.

LEARN MORE

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com