Revolutionizing DAST with IAST: A new era in application security

Introduction

Dynamic Application Security Testing (DAST) has been used to test the security of software for many years. DAST tools operate by simulating attacks on an application from the outside, much like how a hacker might try to find weaknesses.

There is a crucial need for Application Security (AppSec) tools to identify lines of code, application logic and data flows because these elements provide a deeper understanding of how an application works. By analyzing the code, tools can pinpoint exactly where vulnerabilities exist, allowing developers to address the issues directly. This level of insight ensures more accurate detection and reduces the chances of overlooking critical risks.

Understanding application logic is essential because many vulnerabilities arise from how an application processes information or makes decisions. Without examining the logic, tools may miss flaws like missing authorization or authentication checks, which can lead to significant security breaches.

Tracking data flows is also vital, as it shows how sensitive information moves through the system. This helps identify areas where data might be exposed, such as insecure storage or transmission.

Moreover, identifying specific lines of code and understanding application behavior reduces the likelihood of false positives — situations where tools flag harmless issues as vulnerabilities. False positives waste time and resources, as developers must investigate and resolve problems that don't actually exist. Robust AppSec tools that analyze code, logic and data flows provide a more complete and accurate picture, ensuring strong protection while streamlining the development process.

Background: Understanding DAST

There are numerous descriptions of DAST available. The common theme is that DAST is a security testing methodology that involves analyzing an application in its running state. The National Institute of Standards and Technology (NIST) Special Publication 800-53 (Rev. 5), under Security Assessment (SA-11), guideline 8, emphasizes the importance of dynamic testing. This NIST guideline recommends conducting dynamic analysis of the behavior of software components in response to various inputs and conditions, thus highlighting the importance of DAST in AppSec. NIST guidance recognizes that there are many different ways to analyze a running application.

DAST tools operate by simulating attacks on an application from the outside, much like how a hacker might try to find weaknesses.

These "attacks" are carefully crafted to test the application's security by attempting to exploit potential vulnerabilities. After sending these attacks, the tools analyze the HTTP responses — i.e., the messages sent back by the application's server. These responses provide clues about whether the simulated attack was successful. For example, if the server reacts in an unexpected or unsafe way, it might indicate a vulnerability. This process helps identify security risks without accessing the actual application code.

However, DAST tools do not look at the application's internal workings or source code. They rely solely on the application's behavior and server responses to detect problems. This means they often miss deeper vulnerabilities hidden in the code, such as logic flaws, improper error handling or insecure configurations. Without visibility into the internal processes, DAST tools provide a limited perspective, which can result in missed vulnerabilities and an incomplete understanding of the application's security.

Introducing Interactive Application Security Testing (IAST)

In order to address the limitations of traditional DAST, a new way of performing AppSec Testing (AST) has appeared called Interactive Application Security Testing (IAST). IAST is a form of AppSec testing that analyzes applications in a running state but analyzes them from the inside out, rather than from the outside in. IAST uses instrumentation to monitor an application's internal operations, interactions with libraries, connections with backend systems and more, all in real time while the application is being used.

DAST and IAST are not the same thing

IAST and DAST are distinct approaches to AppSec. DAST tests an application from the outside, simulating external attacks and analyzing responses to identify vulnerabilities. It doesn't access the code. Rather, it relies on runtime behavior to find issues — an approach that may miss deeper flaws.

DAST works best on simple monolithic HTTP websites but struggles with modern apps using Application Programming Interfaces (APIs), and it cannot be used with frameworks that use complex protocols or data formats. Scans can take hours or days to complete.

IAST performs continuous analysis, provides instant detailed results in real time, and can be used during development, functional testing and integration tests. IAST only reports vulnerable behavior that occurs in running applications and APIs.

IAST pinpoints vulnerable lines of code, data flows and application logic: Just facts, no guesswork

IAST works inside the application during execution, going beyond what can be detected using the HTTP responses, revealing vulnerabilities in combination with functional tests. It monitors the code, application logic and data flows in real time, providing detailed insights into vulnerabilities and their exact locations. Unlike DAST, IAST offers more accurate results, reduces false positives and identifies issues within the code.

This enhanced visibility allows for more precise identification of vulnerabilities, offering an accuracy that neither traditional DAST nor Static Application Security Testing (SAST) can match. IAST gives developers the best of both worlds — information about the endpoints like DAST and information about the code like SAST — and more: Its accuracy is unprecedented. This combination is the basis for fast and efficient remediation.

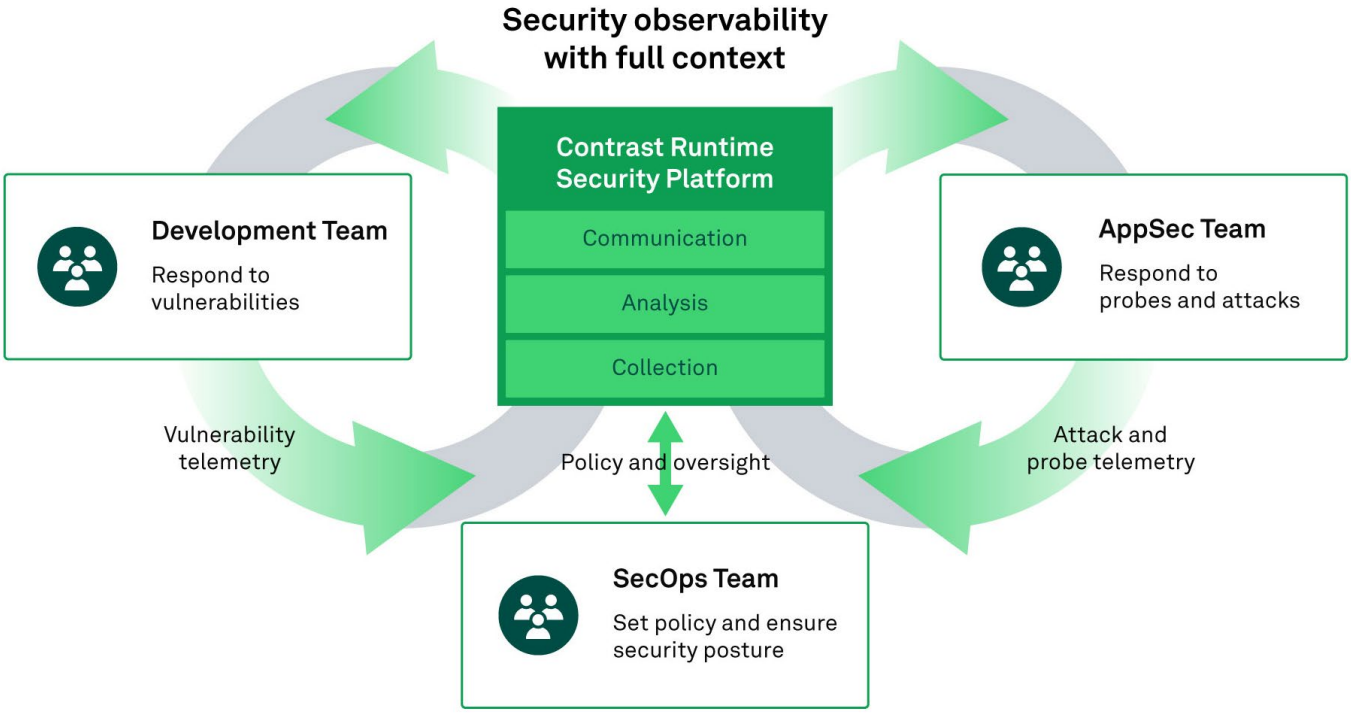


Figure 1: IAST in action: IAST identifies vulnerable code lines and risky data flows, providing full context. The development team addresses vulnerabilities, and the AppSec team monitors and responds to probes and attacks. Meanwhile, SecOps ensures overall security posture, creating a collaborative defense across development, application and operational layers.

IAST takes AppSec testing further by providing unprecedented visibility and accuracy. The context provided by the inside-out, IAST approach enables far superior security testing that provides accurate, actionable results in real time, ensuring robust, comprehensive AppSec. IAST can identify a far broader range of vulnerabilities than DAST, including vulnerabilities like weak encryption that may not be exposed outside the application at all. As well, IAST can only report vulnerable behavior patterns that occur in the running application, ensuring very high accuracy.

Importantly, IAST, unlike traditional DAST, does not require vulnerabilities to be exploited to discover them. Ordinary application traffic, not fuzzing and attack exploits, can be used to find complex vulnerabilities. This opens the world of improved AppSec to anyone, not just experienced AppSec experts. Developers can instantly find vulnerabilities in their code as they do their ordinary quality testing. All Quality Assurance (QA) testing, including automated test cases, can now do double-duty as both QA testing and security testing at once.

Contrast Runtime Security: A new era in AppSec

The need for a comprehensive AppSec testing tool that goes beyond traditional DAST has led to the development of innovative solutions like the Contrast Runtime Security Platform. This approach enables a more accurate and insightful way to pinpoint lines of code, data flows and application logic, providing full context associated with application vulnerabilities.

Contrast works by embedding security instrumentation within the application code. Contrast works by embedding security instrumentation within the application code – an approach pioneered by Contrast. Unlike traditional DAST that analyzes the HTTP response, Contrast moves analysis into the running application. This instrumentation technique has been used for decades in the performance market and is the basis of tools like New Relic, AppDynamics and DataDog.

Unlike traditional DAST tools that have no insight into what happens inside the code, Contrast tracks the control flow, data flow, library use and dangerous functions as the software executes. For example, Contrast can detect when untrusted data is used in dangerous ways, such as being appended directly into a SQL query without the proper defenses being applied. This detection capability is highly accurate and does not require exploiting the vulnerability to discover it.

The Contrast Runtime Security Platform is a novel twist on the DAST model. Like traditional DAST, Contrast analyzes running applications, but it moves the analysis within the code, identifies vulnerabilities more accurately and allows for real-time feedback to developers. Contrast doesn't require any scanning or exploits.

Contrast provides an AppSec dashboard with highly detailed vulnerability descriptions that include the HTTP request, the exact lines of code involved and the exact flow of data through an application or API. Unlike traditional DAST solutions that require a centralized scanner with complex scheduling, Contrast is a distributed approach and can run in parallel across hundreds or thousands of applications, uncovering security flaws in development, pipeline, QA or even production.

With its ability to provide actionable insights into potential vulnerabilities, the Contrast Runtime Security Platform is a powerful tool for developers, helping them to understand and to fix code issues quickly and efficiently. This accelerates the feedback loop, making the AppSec process more efficient and cost effective.

Contrast Runtime Security: A new era in AppSec (cont.)

Contrast's AppSec methodology enables organizations to safeguard their software applications more effectively by building on the strengths of traditional DAST and adding powerful new capabilities. In doing so, Contrast provides a superior approach to ensuring AppSec and meeting DAST requirements.

Moving beyond IAST with AVM: Catch vulnerabilities in production before attack

Contrast Security's ability to identify application vulnerabilities in production environments is a critical advancement in AppSec.

Traditionally, application and API security testing have been done before production, without any insight into real attacks or how software actually runs in production. As a result, development and AppSec teams are drowning in theoretical risk and false positives. By identifying the real, exploitable risks in a running app in production, and enriching them with details about real attacks and exploits, Application Vulnerability Monitoring (AVM) automatically enables teams to focus on the risks that matter, before attackers find them.

Unlike traditional tools that rely solely on pre-deployment testing, Contrast Security provides real-time insights into vulnerabilities as applications operate in live environments. This approach ensures that security is not just a one-time checkpoint but an ongoing process, addressing threats that might emerge after deployment.

By continuously monitoring applications, Contrast Security identifies flaws that may go unnoticed during development, such as configuration issues, insecure APIs or evolving attack vectors. This proactive detection minimizes the window of exposure, reducing the risk of data breaches and system downtime.

Moreover, integrating security into production aligns with modern DevSecOps practices, enabling teams to prioritize and remediate vulnerabilities without disrupting application performance. With Contrast Security, organizations can confidently secure their applications in dynamic, real-world conditions, enhancing both operational resilience and customer trust.

Correlate vulnerabilities with attacks

AVM works within applications to find application and API vulnerabilities in production, correlating these vulnerabilities with real-time attack data from ADR is essential to truly understand risk.

AVM allows organizations to tackle well-known security problems:

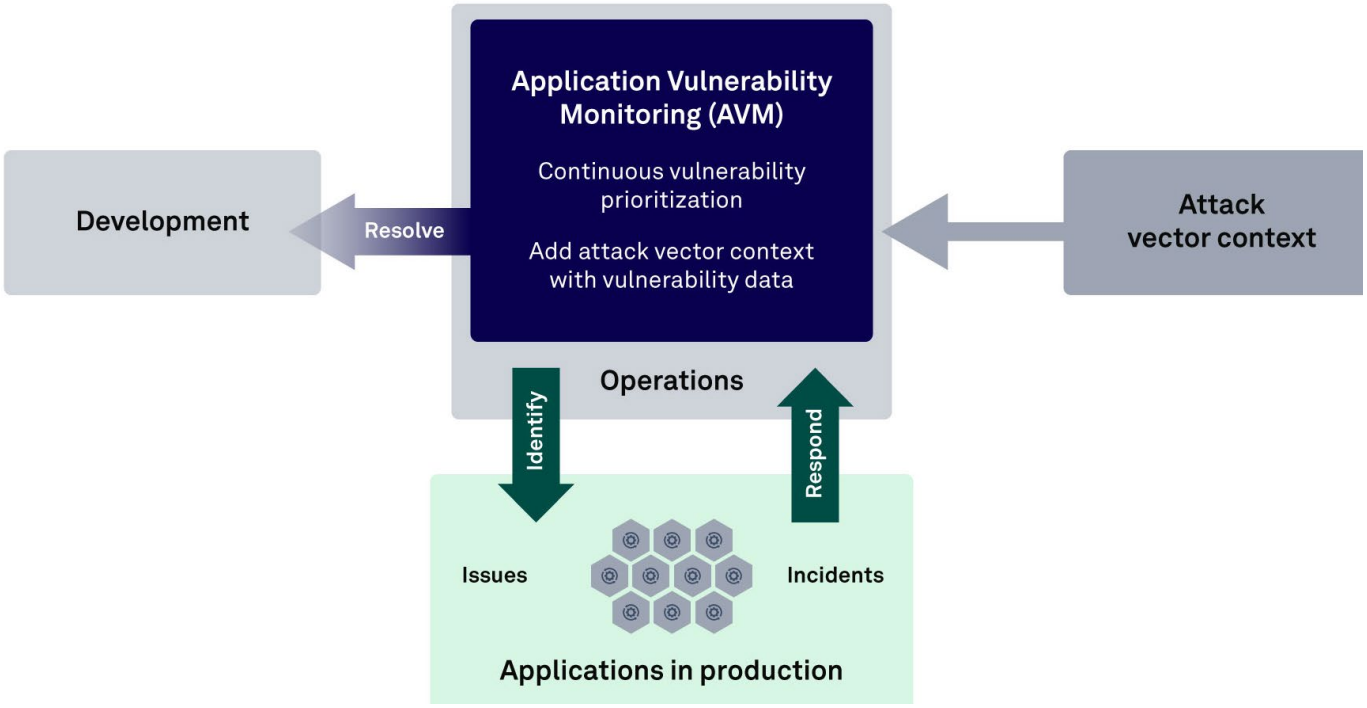
- **Solve for expanding application attack surface:** Organizations using AI to accelerate development often struggle to manage their expanding attack surface. AVM provides continuous visibility within production applications, enabling secure innovation minus the risk.

- **Solve for application risk blindspots:** Organizations struggle to prioritize application vulnerabilities. AVM allows them to see the real exploitable risks in production and what's actually being attacked. This allows SecOps to deploy compensating controls such as Contrast Application Detection and Response (ADR) while developers are implementing a permanent fix.

- **Solve for inefficient incident response:** Organizations can't always identify the vulnerabilities exploited in a security incident because they are using traditional tools. The combination of AVM and ADR can now allow them to rapidly see the entry point, the context surrounding it and the necessary fix.

- **Solve for zero-day attacks:** Organizations are blind to unreported vulnerabilities with traditional approaches. Contrast AVM and ADR works within the application, continuously analyzing behavior and identifying vulnerabilities in real-time, so that organizations can stop and fix issues before they are widely known.

Accurately identifying the issues in production with AVM results in lower overall cyber risk. With AVM, SecOps teams, AppSec teams and DevOps teams can collaborate to prioritize and close exposed vulnerabilities in both custom code and libraries.

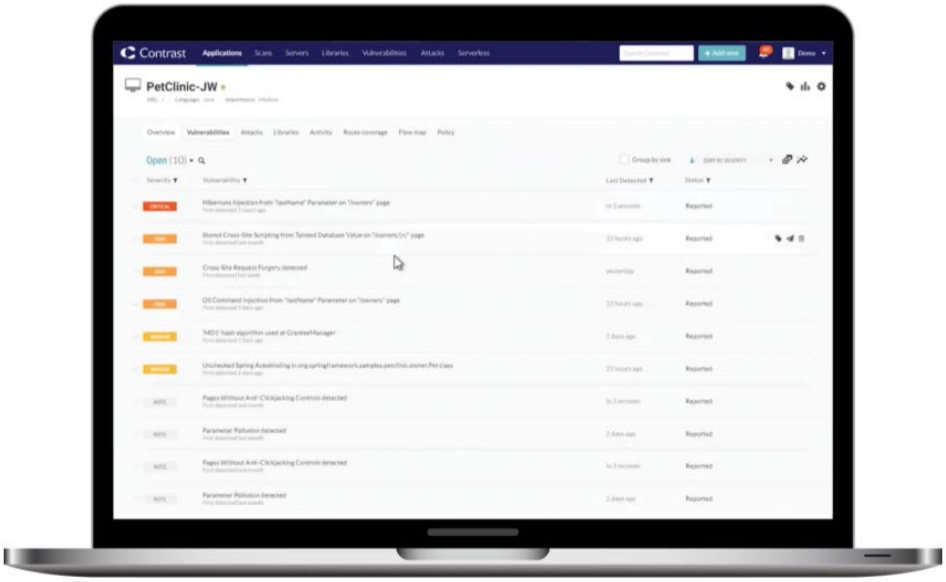


Contrast Runtime Security in action

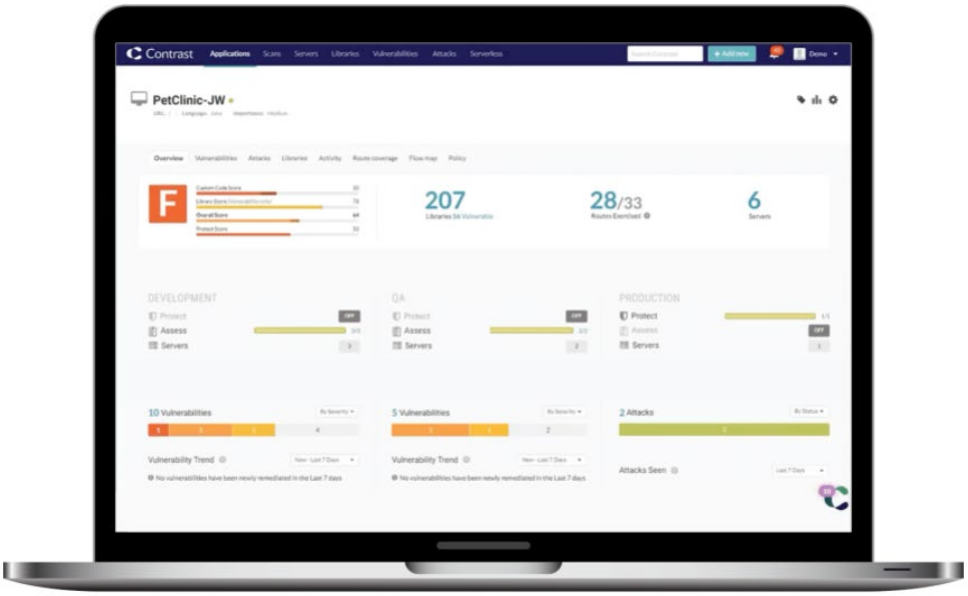
To illustrate the power and efficiency of the Contrast Runtime Security Platform, let's look at it in action using a typical Spring Boot application, Spring Pet Clinic.

The first step involves adding the Contrast Agent to the application stack, which takes a matter of seconds. Once the Contrast agent is added, the application is run in its usual state and starts monitoring the application in the background.

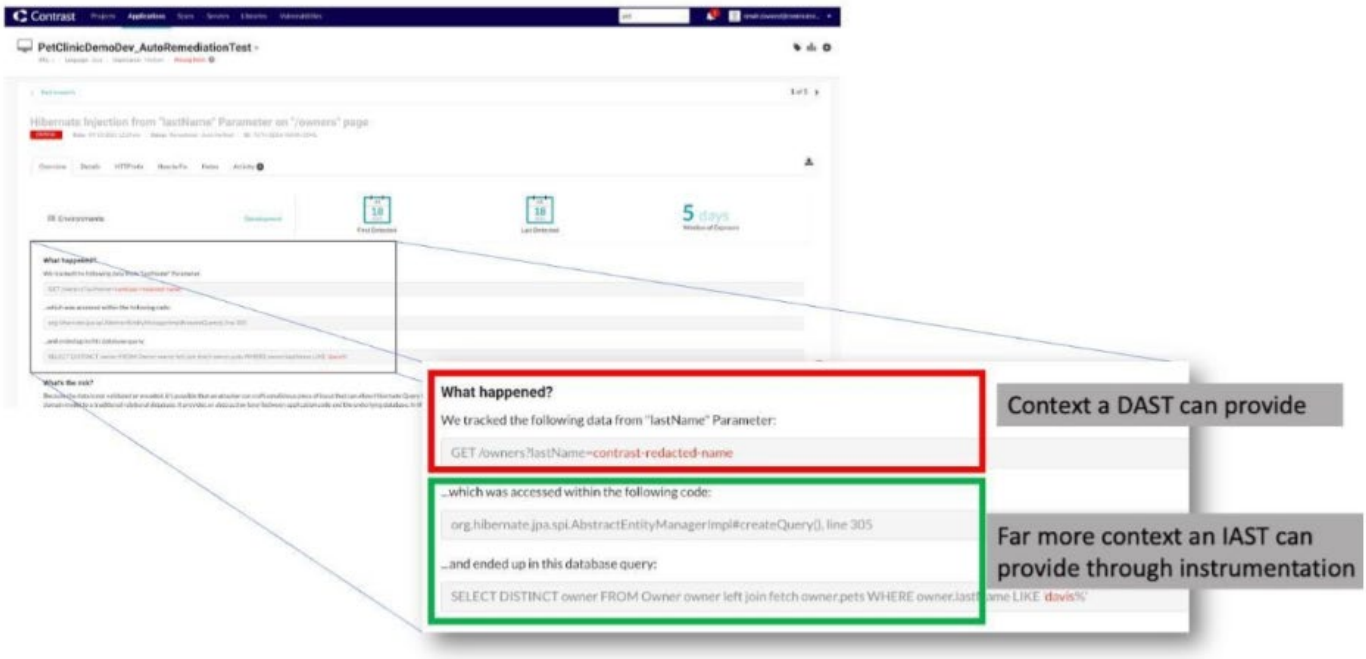
As you use the application, Contrast analyzes each request as it passes through the running code. It detects vulnerabilities and reports them back to the Contrast dashboard.



The dashboard presents an overview of all monitored applications, providing information about the libraries used, routes exercised and servers where the applications run.



When a vulnerability is detected, Contrast provides a detailed breakdown. It shows the tracked data from the HTTP request, the method it ended up in, and how it was used in a query without proper escaping or parameterizing. The lines of code and the entire data flow story are provided, making it easier for developers to understand and fix the vulnerability.



Contrast's ability to simultaneously analyze every input in an HTTP request drastically reduces the time required for security testing. Contrast effectively bridges the gap between traditional DAST and the next generation of AppSec, marking a significant shift in DAST methodology. The high accuracy, immediate feedback and the ability to test multiple attack vectors simultaneously makes the Contrast Runtime Security Platform a powerful tool for AppSec.

How Contrast Security encompasses and surpasses DAST strengths

Contrast Security's solution integrates the strengths of DAST while significantly enhancing its capabilities. Like DAST, Contrast enables detection of externally visible vulnerabilities, ensuring protection against common attack vectors. However, Contrast goes beyond by embedding security testing directly into the application runtime, much like IAST. This allows for real-time vulnerability detection, complete with detailed context such as the exact code lines and vulnerable data flows.

Contrast Security excels by providing continuous, real-time monitoring and feedback, enabling immediate remediation. It reduces false positives by correlating vulnerabilities with runtime data, eliminating unnecessary noise. Additionally, Contrast supports seamless integration into CI/CD pipelines, allowing security to scale with development and operational needs.

How Contrast Security encompasses and surpasses DAST strengths (cont.)

By combining the external testing strengths of DAST with the deep, contextual analysis of IAST, Contrast Security delivers a unified and comprehensive security solution. This empowers development, AppSec and SecOps teams to collaborate effectively, ensuring robust protection while accelerating the delivery of secure software.

Conclusion

In considering the future of AppSec testing, we strongly recommend opting for IAST, such as the Contrast Runtime Security Platform, as it both encompasses the strengths of DAST and goes beyond those strengths. It provides a holistic view of the application, highlighting the vulnerabilities from within and offering superior accuracy and context. Thus, IAST can effectively replace traditional DAST, eliminating the need to implement both, thereby reducing complexity and overhead costs.

6800 Koll Center Parkway,
Ste 235
Pleasanton, CA 94566
Phone: 888.371.1333

Contrast SECURITY