Contrast SECURITY

Differences between active and passive IAST and how to get the best of both worlds

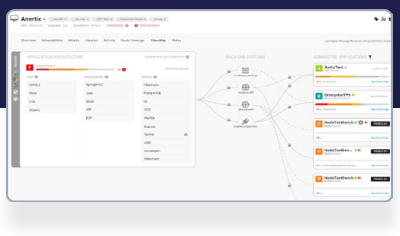
Active and passive AppSec testing

Compared with its predecessors, Interactive Application Security Testing (IAST) is relatively new. The challenge to early adopters is that many technologies and functionalities are bundled under a soup of acronyms. This creates confusion and can be misleading.

This article elaborates on the most popular ways to protect applications from cyberattacks in today's market and how these technologies work, including Contrast's own IAST solution: Assess.

How does Contrast Assess do it?

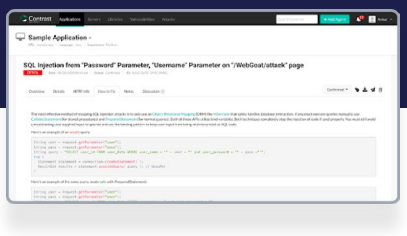
Find and fix the vulnerabilities that really matter — in realtime



1. Instrumentation-based visibility

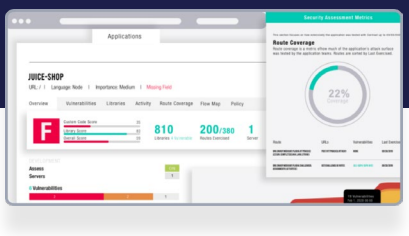
Pinpoint and prioritize software vulnerabilities by embedding sensors inside applications.

Adopting a "shift left" approach to discover vulnerabilities earlier in the Software Development Life Cycle (SDLC).



2. Developer remediation guidance

Learn and identify exactly where a vulnerability appears in the code. Enables developers to fix vulnerabilities easily without the need of security expertise.



3. Accurate results

Test the entire surface of your application by analyzing route and URL traffic to better understand where to effectively increase security test coverage.

WHAT IS IAST?

Analyst firm Gartner has defined the IAST category as follows:

"Interactive Application Security Testing (IAST) uses instrumentation that combines Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST) techniques to increase the accuracy of application security testing. Instrumentation allows DAST-like confirmation of exploit success and SAST-like coverage of the application code and, in some cases, allows security self-testing during general application testing. IAST can be run stand-alone or as part of a larger AST suite, typically DAST."

Gartner's definition is relatively broad, allowing various solutions to be classified as IAST products.

In practical terms, the difference between IAST products is significant. Two relatively new terms have emerged that differentiate the way IASTs fundamentally work.

ACTIVE IAST

This approach requires two main components — a DAST tool and a sensor that attaches to running applications. The advantage of doing it this way instead of running just a DAST scan is that the sensors attached to the application provide additional insight into the exploit compared with the black box nature of typical DAST findings.

During the testing phase, if the application is attacked, active IAST scans the URLs and sends them a list of known attack payloads. The sensor then monitors the application for vulnerabilities based on the incoming attack payloads. Organizations using this approach must still wait for a separate security scan to complete and receive a snapshot of their Application Security (AppSec) status. You can read more about how IAST compares against older predecessors like DAST and the leading advantages of IAST tools as detailed in IDC TechBrief Report 2022.

PASSIVE IAST

Passive IAST is a security tool that requires a single agent to be run alongside an application. It differs from active IAST, which requires actively attacking an application to identify vulnerabilities, as the passive IAST agent continuously monitors all traffic directed at the application at runtime to identify vulnerabilities. The most significant difference here is that organizations no longer need to attack an application to actively find security vulnerabilities.

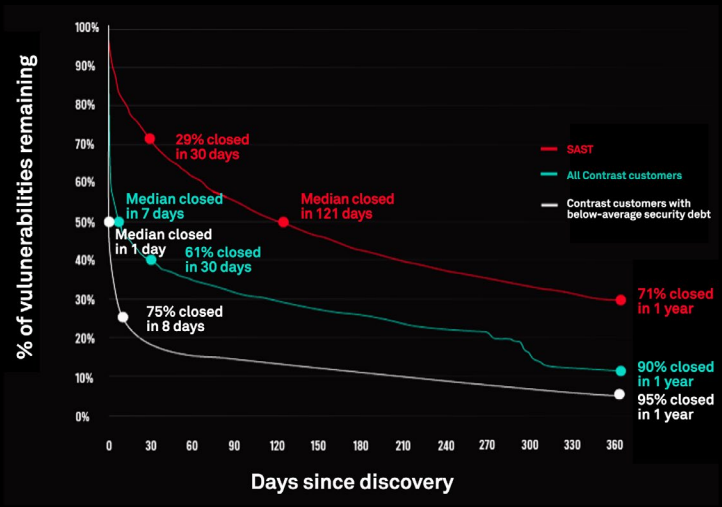
The most comprehensive coverage of an application can be achieved by using existing quality assurance testing — be it manual or automated — or even by testing production use of the application. Passive IAST transforms all use of the application into a security test, making it a cost-effective and secure solution. There is no need to set up a separate infrastructure for security testing.

The passive IAST agent silently and continuously monitors all regular traffic directed at the application to find vulnerabilities at runtime. The main advantage of the passive IAST approach is its cost-effective way of finding more security issues in an application, as it can detect both known and unknown vulnerabilities without the requirements of security experts or additional tools.

The Contrast difference

45x faster remediation with Contrast's modern security platform

Note: SAST results sourced from
Veracode State of Software Security Report 2020



OWASP Benchmark

Free and open application security test suite with 2,740 security challenges

Source: OWASP Benchmark Project V1.1 & V1.2, 2016

WHICH APPROACH IS BETTER FOR ME?

Active and passive IAST each has its advantages. For example, if an application is extremely immature in its testing, active IAST can still find vulnerabilities. However, in the majority of cases, it will be the passive approach that will be the most scalable and manageable across AppSec programs.

A passive IAST approach has the following advantages

- It does NOT require you to attack the application and instead finds vulnerabilities through regular traffic flowing through the app;
- It does NOT take an extended period of time to scan in your Continuous Integration/Continuous Deployment (CI/CD) pipeline and instead monitors your application in realtime;
- Active IAST is a point-in-time snapshot that is only applicable in the testing phase, as opposed to passive IAST, which fits itself across your entire SDLC;
- Passive IAST incentivizes you to improve your application testing coverage, as you now get twice the value from it (code quality as well as code security); and
- Finally, passive IAST has fewer moving parts, making it easier to manage and scale across many applications.

HOW DOES CONTRAST ASSESS ENABLE THE IDEAL IAST OUTCOMES FROM THE BEST OF BOTH WORLDS?

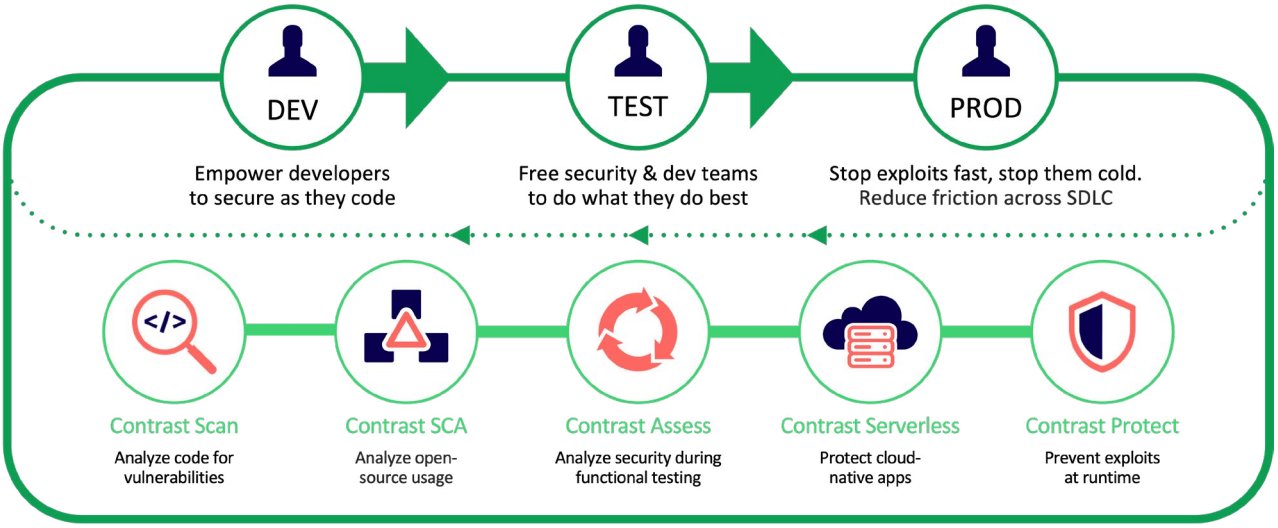
Contrast Assess enables users to achieve the best of both outcomes from passive and active IAST. From the outset, Assess is primarily a passive IAST solution that continuously monitors your application through regular traffic that passes through. If you want to implement active IAST, your best approach would be to use a passive IAST and choose an automated testing tool that acts like a crawler — or even a free one like OWASP Zed Attack Proxy (ZAP).

You'll generate better results by generating simple end-to-end tests with tools that don't require vulnerabilities to be exploited.

In summary, passive IAST is the ideal approach for most applications. It is scalable and offers the best value for money. Active IAST should be used in combination with passive IAST to detect vulnerabilities in immature applications. Contrast Assess enables users to achieve the best of both outcomes and perform security self-testing during general testing. This helps to ensure the security of applications before they are released.

The Contrast Secure Code Platform

One solution, dev-through-prod protection.



Contrast Security provides the industry's most modern and comprehensive Application Security Platform, removing security roadblocks inefficiencies and empowering enterprises to write and release secure application code faster. Embedding code analysis and attack prevention directly into software with instrumentation, the Contrast platform automatically detects vulnerabilities while developers write code, eliminates false positives, and provides context-specific how-to-fix guidance for easy and fast vulnerability remediation. Doing so enables application and development teams to collaborate more effectively and to innovate faster while accelerating digital transformation initiatives. This is why a growing number of the world's largest private and public sector organizations rely on Contrast to secure their applications in development and extend protection in production.

240 3rd Street
2nd Floor
Los Altos, CA 94022
Phone: 888.371.1333
Fax: 650.397.4133

Contrast SECURITY

contrastsecurity.com