

Hdiv Detection (IAST)

<h2>Higher Accuracy</h2> <p>Hdiv IAST scores 100% in the OWASP Benchmark test, with no false positives, compared to around 80% for the best SAST.</p> <p>Interactive Application Security Testing (IAST) tools <u>combine the static approach and the dynamic approach</u>.</p> <p>There is no need to have access to the source code of the application, and there is no need to attack the applications.</p>	<div><div><div><div><div>TRUE POSITIVE</div><div>(Perfect score: 100%, Worst score: 0%)</div></div><div><div><div>18%</div><div>Best DAST</div></div><div><div>84%</div><div>Best SAST</div></div><div><div>100%</div><div>Hdiv IAST</div></div></div><div><div><div>OWASP</div><div>Open Web Application Security Project</div><div>Benchmark</div></div></div></div><div><div><div>FALSE POSITIVE</div><div>(Perfect score: 0%, Worst score: 100%)</div></div><div><div><div>1%</div><div>Best DAST</div></div><div><div>53%</div><div>Best SAST</div></div><div><div>0%</div><div>Hdiv IAST</div></div></div></div></div></div>		
<h2>Applicable to All Stages of the SDLC and to All Environments</h2> <ul style="list-style-type: none">• Development: incorporate detection of application security issues from the first day, with no false positive distractions• QA: introduce automatic security checkpoints and support of CI/CD pipelines (integrates with Jenkins, Maven, and others)• Production: monitor application vulnerabilities in real time, just by leveraging regular traffic (no need to attack the applications) and option to upgrade to RASP Protection of the vulnerabilities.	✗	✗	✓
<h2>Continuous Vulnerability Detection</h2> <p>No need to break the workflow of developers, QA teams, and security teams to perform a lengthy manual scan of the application. There is no need to attack the application. Vulnerability lists are always up-to-date, even when the applications change continuously. Everything is automatic.</p>	✗	✗	✓
<h2>Third-party component vulnerability detection</h2> <p>Even though SAST tools identify vulnerable libraries, they often struggle to incorporate new architectures such as APIs and REST endpoints because their approach is based on static analysis. IAST tools, on the other hand, follow the runtime execution flow, including third party components vulnerability detection.</p>	✗	✗	✓

Hdiv Detection (IAST)

Application Server instrumentation via an agent

Development

Install Hdiv agent locally in the developers' machines to start security checks from the first day



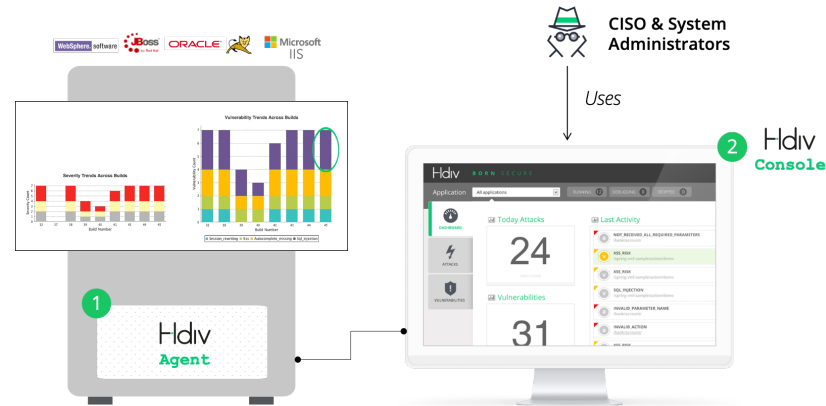
1 Agent

<https://hdivsecurity.com/videos/hdiv-detection-ia-st-installation>

REAL-TIME VISIBILITY

QA

Automatically stop build pipelines when certain customizable security thresholds are not met (such as the number and severity of application security vulnerabilities)



1 Agent

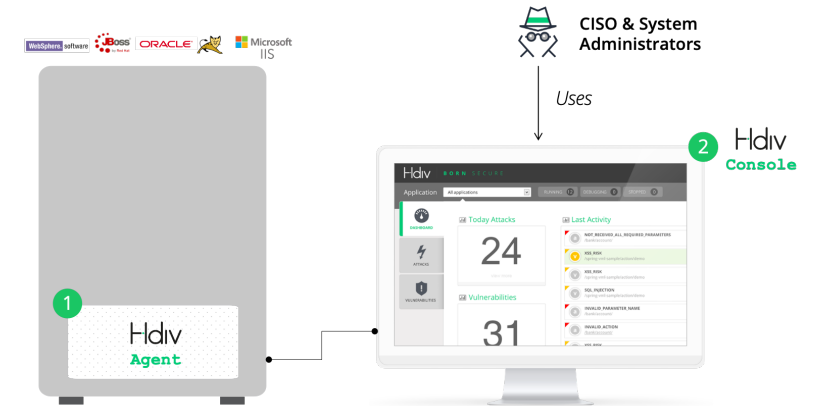
2 Admin & Management Console

BUILD AND CI/CD INTEGRATIONS

maven  **Jenkins**

Production













Security teams benefit from constant visibility of the security position of the applications; no need to perform audits nor manually attack the applications. Upgrade to protection of the existing vulnerabilities.



1 Agent

2 Admin & Management Console


REAL-TIME VISIBILITY
PREVENTION OF THE EXPLOITATION OF VULNERABILITIES

	SAST	DAST	IAST	RASP
Timeline	Development	Testing, Production	Development, Testing, QA, Production	Production
Speed	Instant to hours	Hours to days	Instant (at runtime)	Instant (at runtime)
How it works	Analyzes static code to identify vulnerabilities	Sends HTTP requests to test behavior of web apps	Analyzes code and behavior of running apps through instrumentation	Monitors and protects apps at the runtime or server layer
Allows continuous security testing				
CI/CD integration				
Accuracy True Positive: <ul style="list-style-type: none"> Perfect score: 100% Worst score: 0% False Positive: <ul style="list-style-type: none"> Perfect score: 0% Worst score: 100% 	OWASP Benchmark results: <ul style="list-style-type: none"> True Positive: 84%  False Positive: 53%  	OWASP Benchmark results: <ul style="list-style-type: none"> True Positive: 18%  False Positive: 1%  	OWASP Benchmark results: <ul style="list-style-type: none"> True Positive: 100% False Positive: 0% 	High
Actionability	High: points to vulnerable lines of code	Low: difficult to deduce location of problem	High: points to vulnerable lines of code	High: detailed information on attacks

Vulnerability Coverage

- ✓ Cross-Site Scripting
 - ✓ Command Injection
 - ✓ Unvalidated Redirects
 - ✓ Reflection Injection
 - ✓ Log Injection
 - ✓ X-Content-Type Header Missing
 - ✓ Weak Browser Cache Management
 - ✓ Autocomplete Missing
 - ✓ Directory Listing Leak
 - ✓ SQL Injection
 - ✓ Header Injection
 - ✓ Insecure Transport Protocol
 - ✓ Weak Protocol
 - ✓ LDAP Injection
 - ✓ HSTS Header Missing
 - ✓ Weak Randomness
 - ✓ Insecure Cipher
 - ✓ Session Timeout
 - ✓ CSP Header Missing
 - ✓ Insecure Cookie
 - ✓ PCI Logging Violation
 - ✓ Trust Boundary Violation
 - ✓ HTTP Method Tampering
 - ✓ Clickjacking Control Missing
 - ✓ Insecure Hashing
 - ✓ Insecure JSP Layout
 - ✓ XPath Injection
 - ✓ **NEW Magecart Detection**
- ...and more





Favor AST with fast turnaround and low false positives.
Tune AST to reduce false positives, and consider **IAST**
when **speed and accuracy** are paramount, especially in
DevSecOps initiatives.



CISO PLAYBOOK: EMBEDDING AST IN THE SOFTWARE DEVELOPMENT LIFE CYCLE, BY AYAL TIROSH (NOVEMBER 2017)