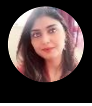

Rashmi Bhardwaj

SAST vs DAST vs IAST – Cheat Sheet

Feature	SAST Static Application Security Testing	DAST Dynamic Application Security Testing	IAST Interactive Application Security Testing
Definition	Analyzes source code, bytecode, or binaries for security vulnerabilities without executing the application.	Tests an application during runtime by simulating attacks from an external perspective.	Uses agents or instrumentation to analyze applications during runtime.
Testing Phase	Early in the Software Development Life Cycle (SDLC)	Late-stage or post-deployment testing	During runtime but in pre-production or testing environments
Execution Type	Static (without running the application)	Dynamic (application must be running)	Hybrid (both static and dynamic analysis)
Focus Area	Source code, configuration, and dependencies	HTTP/HTTPS traffic, APIs, and runtime interactions	Real-time application behavior and runtime execution
Detection Capabilities	Finds security vulnerabilities in code before deployment.	Identifies vulnerabilities that are exploitable in a running application.	Detects real-time vulnerabilities based on actual execution paths.
Commonly Found Issues	SQL injection, XSS, insecure coding practices, hardcoded secrets	Injection attacks, authentication issues, security misconfigurations, API vulnerabilities	Runtime injection attacks, unauthorized access, business logic flaws
False Positives	High (since it doesn’t verify exploitability)	Moderate (depends on the accuracy of simulation)	Low (since it sees actual execution paths)
False Negatives	Low (finds issues in code that might not be executed)	Higher (might miss vulnerabilities in unexecuted parts of the application)	Low (monitors both execution flow and vulnerabilities)
Ease of Integration	Requires access to source code and integrates into CI/CD pipelines	Does not require source code, can be used on deployed applications	Requires both testing and running environments
Performance Impact	No impact on application performance	Can slow down the application during testing	Some impact, but mostly in testing environments
Compliance Support	Helps meet security compliance early in SDLC	Supports compliance by detecting runtime vulnerabilities	Aids in compliance by validating security at runtime
Best Used For	Development phase security assessments	QA and security testing before deployment	Advanced vulnerability detection during runtime testing
Examples of Tools	Checkmarx, Fortify, SonarQube, Veracode	OWASP ZAP, Burp Suite, Acunetix, AppScan	Contrast Security, Seeker, Hdiv
Primary Users	Developers, security teams	Security testers, QA teams	DevSecOps, security analysts