Jason Luttrell

CSD 370

Module 5.2: **Security Testing**

**DAST (Dynamic Application Security Testing)**

**Description:** DAST scans a running application from the *outside* by sending malicious inputs and analyzing responses, looking for security vulnerabilities like XSS or SQL injection — like a hacker would. It is like automated Pen Testing.  
**Tool:** *OWASP ZAP* — an open-source tool that automatically scans web apps for vulnerabilities during runtime. (OWASP ZAP, n.d.)  
**When:** Typically used during the **testing phase**, once you have a working build deployed to a test environment.

**SAST (Static Application Security Testing)**

**Description:** SAST analyzes your source code, bytecode, or binaries *without running the program*, catching security flaws like insecure coding patterns or input validation issues. It is like a really good automated code review.  
**Tool:** *SonarQube* — an automated static analysis tool that finds bugs, code smells, and security vulnerabilities in your source code. (SonarQube, n.d.)  
**When:** Used **early in development**, ideally during coding, as part of code reviews or in a CI pipeline.

**IAST (Interactive Application Security Testing)**

**Description:** IAST combines aspects of DAST and SAST — it monitors a running application *from inside* during functional testing, providing real-time vulnerability detection with detailed code-level context.  
**Tool:** *Contrast Security* — an IAST tool that instruments your app to analyze runtime data flows and detect vulnerabilities in real-time. (Contrast Assess Documentation, n.d.)  
**When:** Best during **integration and system testing**, when the application can be exercised with real traffic.

**RASP (Runtime Application Self-Protection)**

**Description:** RASP runs inside your application and automatically detects and blocks attacks in real time by monitoring app behavior and context.  
**Tool:** *Imperva RASP* — a runtime security tool that blocks exploits automatically without needing network perimeter rules. (Imperva RASP, n.d.)  
**When:** Implemented in **production**, protecting live applications while they run.

**References**

OWASP Foundation. (n.d.). *OWASP ZAP Documentation*. https://www.zaproxy.org/docs/

SonarSource. (n.d.). *SonarQube Documentation*. https://docs.sonarsource.com/sonarqube/latest/

Contrast Security. (n.d.). *Contrast Assess Documentation*. https://docs.contrastsecurity.com/en/contrast-assess.html

Imperva. (n.d.). *Runtime Application Self-Protection (RASP)*. https://docs.imperva.com/bundle/v14.7-rn/page/88726.htm