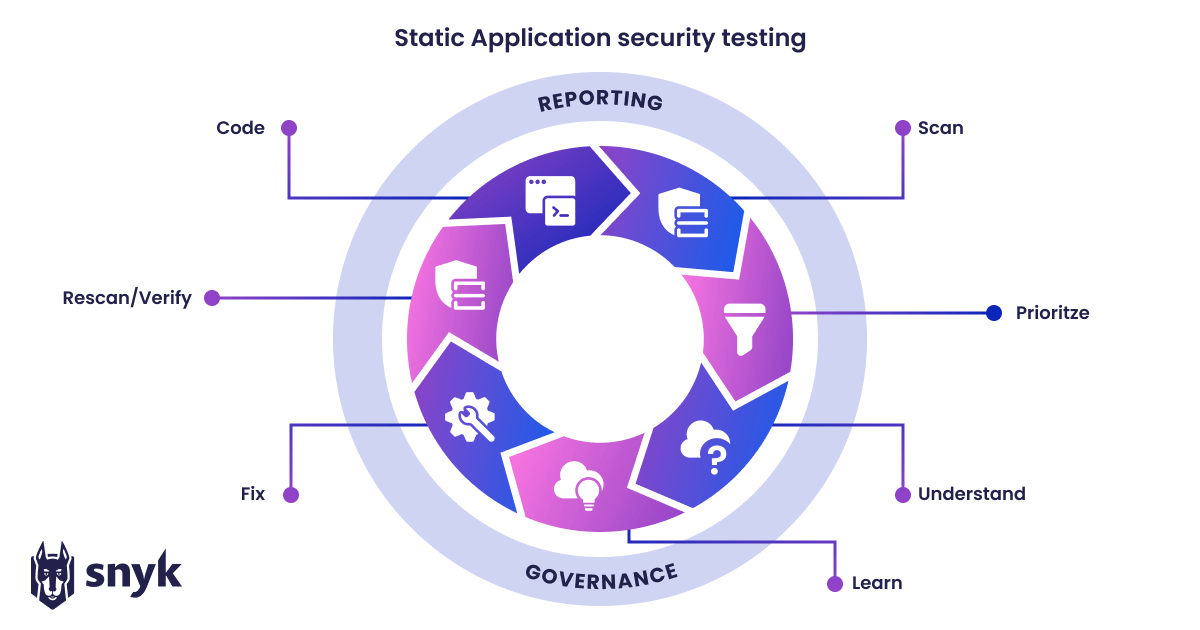
**Case Study 4**

**Aim** - To study static application security testing (SAST) tools.

**Theory -**

Static Application Security Testing (SAST) is a type of security testing that analyzes source code, bytecode, or binary code to identify security vulnerabilities, coding flaws, and other weaknesses in an application. SAST tools work by examining the application's codebase without actually executing the application. They help developers identify potential security issues early in the development lifecycle, allowing for timely remediation.

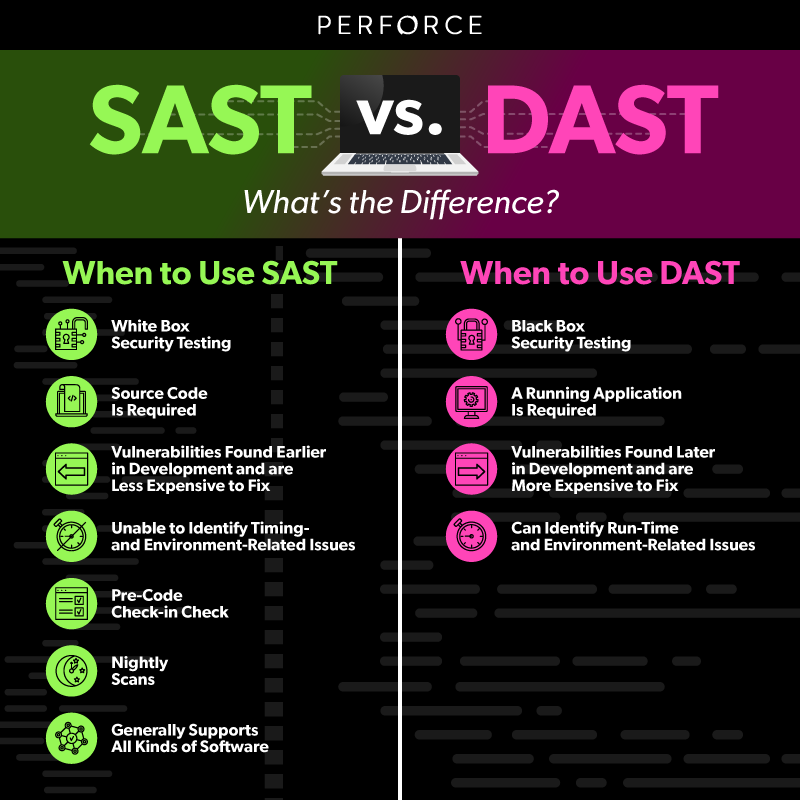


**1. What is the primary purpose of using Static Application Security Testing (SAST) tools in software development?**

Answer: The primary purpose of using SAST tools is to identify security vulnerabilities and coding flaws in the application's source code, bytecode, or binary code. This helps developers detect and address potential security issues early in the development process.

**2. How does Static Application Security Testing (SAST) differ from other types of security testing, such as Dynamic Application Security Testing (DAST)?**

Answer: SAST focuses on analyzing the application's code without executing it, while DAST involves testing the application in a running state. SAST tools scan for coding flaws and vulnerabilities within the codebase, whereas DAST tools simulate real-world attacks on the deployed application.



**3. What are some common types of security vulnerabilities that Static Application Security Testing (SAST) tools can help identify?**

Answer: SAST tools can identify a wide range of vulnerabilities, including SQL injection, Cross-Site Scripting (XSS), code injection, insecure authentication mechanisms, and more. They can also detect coding practices that might lead to vulnerabilities, such as buffer overflows or insecure data handling.

**4. How do Static Application Security Testing (SAST) tools contribute to the overall software development process?**

Answer: SAST tools contribute to the software development process by enabling developers to find and fix security vulnerabilities during the coding phase. This proactive approach helps prevent vulnerabilities from reaching the production environment, reducing the risk of security breaches and the associated costs of addressing issues post-deployment.

**Conclusion -**

In conclusion, static application security testing (SAST) tools are essential components of a comprehensive software security strategy. They play a vital role in identifying and addressing potential security vulnerabilities and coding flaws before the application is deployed. By incorporating SAST tools into the development process, organizations can enhance the overall security posture of their software applications and reduce the risk of security breaches.