**EXPERIMENT 04**

**AIM:** Study of SAST Tools.

**TO DO:**

1. What is SAST ?
2. Importance
3. Limitations
4. SAST Tools

**THEORY:**

**What is SAST ?**

Static Application Security Testing (SAST) is a frequently used Application Security (AppSec) tool, which scans an application’s source, binary, or byte code. A white-box testing tool, it identifies the root cause of vulnerabilities and helps remediate the underlying security flaws. SAST solutions analyze an application from the “inside out” and do not read a running system to perform a scan.

SAST reduces security risks in applications by providing immediate feedback to developers on issues introduced into code during development. It helps educate developers about security while they work, providing them with real-time access to recommendations and line-of-code navigation, which allows for faster vulnerability discovery and collaborative auditing. This enables developers to create more code that is less vulnerable to compromise, which leads to a more secure application, and less need for constant updates and modernization of apps and software.

**Importance**

SAST is an essential step in the Software Development Life Cycle (SDLC).

1. It identifies critical vulnerabilities in an application before it’s deployed to the public, while they’re the least expensive to remediate. It’s in this stage of static code analysis that developers can code, test, revise, and test again to ensure that the final app functions as expected, without any vulnerabilities. When SAST is included as part of the Continuous Integration/Continuous Development (CI/CD) pipeline, this is referred to as “Secure DevOps,” or “DevSecOps.”GitHub SAST Integration.
2. SAST allows developers to identify potential security vulnerabilities in the source code at an early stage of the development process. This early detection enables teams to address security issues before they become more complex and costly to fix in later stages.
3. Fixing security issues in the early development stages is generally less expensive and time-consuming compared to addressing them after the application has been deployed. SAST helps reduce the cost of security maintenance by catching issues before they reach production.
4. SAST tools can be integrated directly into the development workflow and Continuous Integration/Continuous Deployment (CI/CD) pipelines. This makes it easier for developers to run security scans regularly, fostering a security-first mindset within the development team.

Analysis shows that 94% of the applications on the web contain bugs in security features, while code quality and API abuse issues have roughly doubled over the past 4 years. If these vulnerabilities are left unchecked and the app is deployed as such, this could lead to a data breach, resulting in major financial loss and damage to your brand reputation.

**Limitations**

1. **Not capable of identifying vulnerabilities in dynamic environments**

SAST tools analyze the application's source code without executing it, which means they lack visibility into the runtime behavior of the application.

Dynamic environments involve interactions with various external components, data sources, user inputs, and other systems, which can lead to different code paths and behaviors at runtime. Since SAST tools only analyze the static code, they might miss security vulnerabilities that only manifest during runtime or are dependent on specific data input and runtime conditions.

1. **High risk of reporting false positives**

Due to the static nature of the analysis, SAST tools can produce a significant number of false positives. This can lead to extra effort for developers in manually verifying and filtering the results.

1. **Limited Context Awareness**

SAST tools analyze the source code in isolation, which means they may not fully understand the application's runtime context. As a result, false positives or false negatives can occur, leading to inaccuracies in the reported security issues.

1. **Incomplete Code Coverage**

SAST tools may not be able to analyze all parts of the codebase, especially when certain code paths are dynamically generated or not reachable during static analysis. This can leave blind spots in the security assessment.

1. **Time-Consuming Scans**

Large codebases may take a long time to analyze with SAST tools, leading to slower feedback cycles for developers, impacting productivity.

1. **Limited Detection of Logic-Based Vulnerabilities**

SAST tools primarily focus on pattern-matching and may miss logic-based vulnerabilities where the flaw is dependent on a series of events or conditions.

**SAST Tools:**

1. **Klocwork**

Klocwork works with C, C#, C++, and Java codebases and is designed to scale with any size project. The static analysis nature of Klocwork works on the fly along with your code linters and other IDE error checkers. It is especially good at finding div by zero, null pointer issues, array out of bounds, and the likes, without running the code to test it.

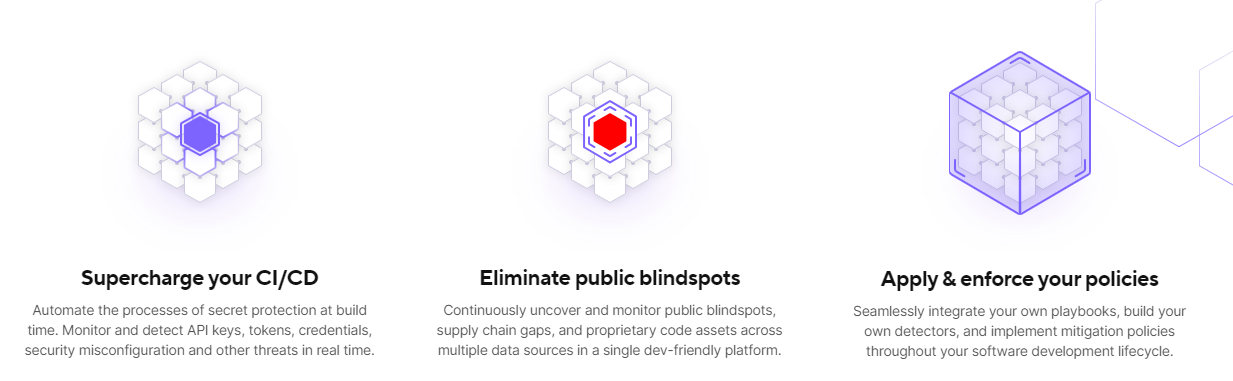
Klocwork can help you adhere to several coding and security standards: CWE, OWASP, CERT, PCI DSS, DISA STIG, and ISO/IEC TS 17961. Users may also add custom checks, although some users found the lack of documentation around the area difficult to maneuver. Klocwork can do pre- and post-check-in analysis as part of your CI/CD pipeline to increase the overall quality of your code.



1. **SpectralOps**

SpectralOps is unique in the landscape since it scans the entire SDLC for hard coded secrets, keys, and misconfigured code, continuously. Spectral is a multi-language AI-driven SAST. The primary objective of Spectral is to prevent secrets (credentials, API keys, encryption keys, etc) from leaking.

Unlike most SAST, SpectralOps avoids false positives by using a sophisticated AI. Avoiding false positives is one of the most important aspects of any SAST, as a high volume of false positives is like your SAST crying wolf. Eventually, developers will ignore the warnings. Secret scanners are an essential part of any security stack you should not overlook.



1. **GitHub Advanced Security**

GitHub Advanced Security (formerly known as GitHub Code Scanning) is a feature provided by GitHub to enhance the security of software development projects. It incorporates Static Application Security Testing (SAST) as one of its key components.

SAST tools examine the code for common security flaws like SQL injection, Cross-Site Scripting (XSS), code injections, and more.

By analyzing the code statically, SAST can identify vulnerabilities early in the development process, reducing the risk of deploying insecure code.

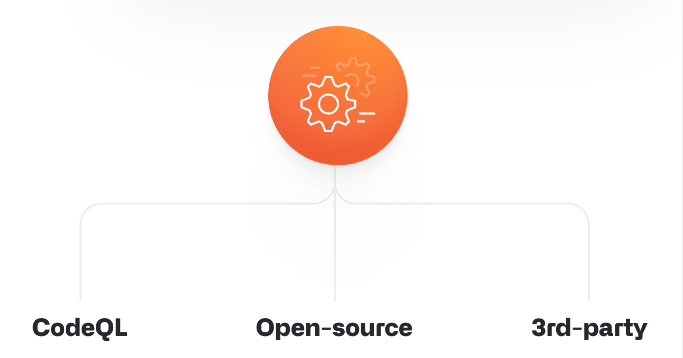
GitHub Advanced Security provides an integrated platform for conducting security analysis within the GitHub repository.

It allows developers to automatically trigger SAST scans during pull requests and code pushes.

SAST scans are conducted using security analysis tools that GitHub supports or tools that can be integrated through GitHub Actions or other CI/CD pipelines.

GitHub Advanced Security provides detailed security insights, including a summary of security issues, their severity, and suggested remediation actions.

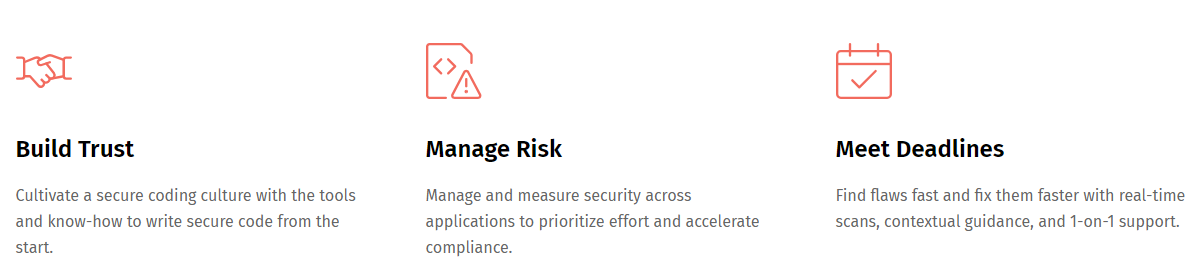
Developers can view the code lines associated with each security issue and gain contextual information to better understand and fix the problems.



1. **Veracode**

Veracode has many security-related software solutions. Their SAST Veracode Static Analysis has a low false-positive count and offers developers potential solutions to issues it finds.

Being Software as a Service means low setup overhead and a quick turnaround from first acquiring access and getting results. However, Veracode does not offer a free version to try out. That said, reviewers are overall pleased with the product, particularly in maintaining security standards.



1. **Codacy**

Codacy is another automated code review tool. Rather than focus on fixing your code for you; the focus is on giving you information about the overall health of your project. When working in a team, it is vital to keep track of technical debt, readability, and adherence to standards. This tool helps you keep track of many different statistics regarding your project.



**CONCLUSION:**

We have studied different SAST Tools and how they help in the overall betterment of applications.