# Above 80% of source codes have security vulnerabilities, helping hackers, while SecureClaw's SAST can avoid such issues

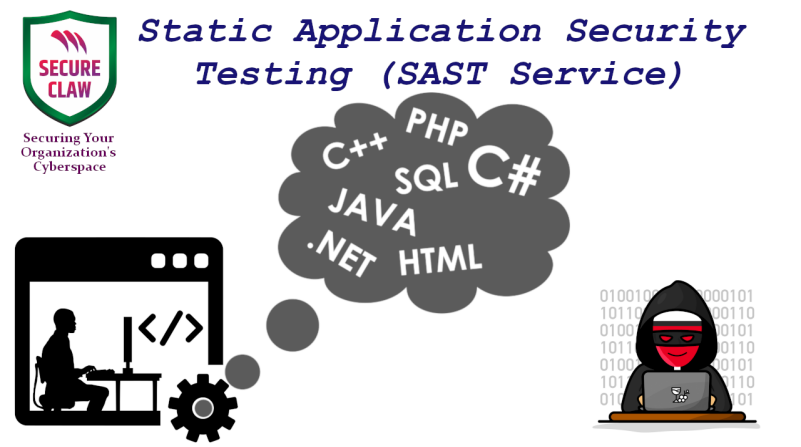
*A good software application provides ease of functionality and faster operations without ignoring the privacy, integrity, confidentiality, and availability.*

Static Application Security Testing (SAST) Service by SecureClaw

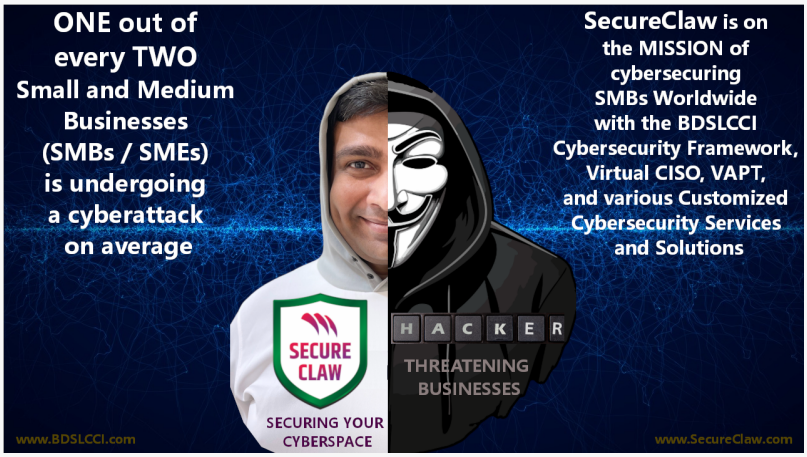
DOVER, DELAWARE, UNITED STATES, November 26, 2023 / EINPresswire.com/ -- According to a Forbes, 84% of source code bases contain open-source vulnerabilities. Each programming language has a unique type of source code, which is sometimes just called "source." This code is written in a text format that is easy for human developers or software engineers to read and contains instructions on what has to be done by a machine. Before being used, some code is compiled, which turns the source code into a collection of machine-language instructions. There are two types of source ownership and distribution in law. “Open-source software” is code that is made available, either freely or with restrictions under an open-source license that protects the author's fundamental rights. Certain agreements restrict the types of changes that can be made to the source code, while others only require that the original author be credited. The other significant legal type of source code is called “closed source”. Under this paradigm, a license holder receives merely an executable file. It is forbidden for users to try to extract the underlying code by decompiling the executable files. The commercial software industry is dominated by this legal model, while several open-source business models have also shown to be successful.

Static Application Security Testing (SAST) is usually performed as part of a Source Code Review (also known as white-box security testing) and is carried out at the Implementation phase of a Security Development Lifecycle (SDLC) or even in Agile Software Development. Static Code Analysis commonly refers to the running of Static Code Analysis tools that attempt to highlight possible vulnerabilities within ‘static’ (non-running) source code by using techniques such as Taint Analysis and Data Flow Analysis.

 Critical Areas considered in Static Application Security Testing (SAST): Since it's a white box testing technique, it needs access to the source code in order to work. By analyzing code before it is distributed, it discovers all security vulnerabilities, including bugs and weaknesses in software like SQL injection and others. To conduct evaluations, SAST does not need a system to be in operation. One very scalable security testing technique is Static Application Security Testing (SAST). It can also be automated, which will help company save money and time. Because SAST testing is done early in the Software Development Life Cycle (SDLC), it is simple to identify possible security flaws sooner.



Source Code Security Review Service by SecureClaw Inc.



SecureClaw Cybersecurity Services

A source code security review centres on around seven critical areas or security factors. Malicious users may find it easy to target any program that does not meet protection requirements in any of these specific areas.

Furthermore, a source code review illustrates the strength of applications in these specific areas and aids the development team in identifying gaps.

- Authorization
- Authentication
- Data Validation
- Session management
- Error handling
- Logging
- Encryption

 Source Code Security Review Process:

The Open Worldwide Application Security Project (OWASP) is an online community that produces freely available articles, methodologies, documentation, tools, and technologies in the field of web application security. The OWASP Foundation provides a list of source code analysis tools that can help analyze source code or compiled versions of code to help find security flaws. These tools can be added into organization's IDE and can help organization detect issues during software development.

> Good software programmer looks both ways before crossing a one-way street of delivery: first, the source code should be refactored, and second, it should follow cybersecurity best practices by design."
>
> *Dr. Shekhar Pawar, CEO, SecureClaw Inc., Inventor of BDSLCCI*

SAST tool feedback can save time and effort, especially when compared to finding vulnerabilities later in the development cycle. However, it's important to note that current SAST tools are limited. They can automatically identify only a relatively small percentage of application security flaws and frequently have high numbers of false positives. Additionally, many SAST tools have difficulty analyzing code that can't be compiled.

Some tools are starting to move into the Integrated Development Environment (IDE). For the types of problems that can be detected during the software development phase itself, this is a powerful phase within the development lifecycle to employ such tools, as it provides immediate feedback to the developer on issues, they might be introducing into the code during code development itself. This immediate feedback is very useful as compared to finding vulnerabilities much later in the development cycle.

Using the combined capabilities of manual and automated reviews is the current industry best practice. Consequently, the optimal method would be able to extract the most sophisticated attack vectors from organization's application's source code.

"When exactly do organization need a source code review?" is the hot topic of the moment. Applications need a code review in the early stages of the software development lifecycle to get the best outcome thus far. Furthermore, it may undoubtedly help developers enhance the application's preparedness in every way possible and quickly fix live vulnerabilities in the codebase. Let's now briefly examine the scenario in which an automatic review can help a developer who is producing code quickly incorporate changes to the codebase in parallel. Additionally, an automated review enables the developer to quickly analyze larger codebases with commercial or open-source tools. Moreover, the SAST tools are used by the avantgarde development team to patch exposure in real time. Conversely, a manual review is helpful when the project is about to be committed. It also includes developer objectives and presumes business logic. To make this procedure clear, which entails a senior code review specialist carefully examining the source code.

In conclusion, while it takes some time, this is sufficient for identifying flaws or problems in enterprise logic.

 Advantages of a Source Code Security Review
The following are the main advantages of carrying out a SAST.
  Reducing the quantity of risks that arise during the last phases of the software development

lifecycle.
- Minimizing the number of security flaws that are currently in the production system.
- Reduce the amount of time developers spend troubleshooting problems, which will increase productivity.
- Improving uniformity, continuity, and maintainability throughout the codebase.
- Improve ROI by using available resources and time to expedite and secure the process.
- Increasing developer learning and productivity to support code development in the future.

SecureClaw's SAST service is helping many organizations worldwide.

Dr. Shekhar Pawar
SecureClaw Inc.
+1 218-718-2121
customercare@secureclaw.com
Visit us on social media:
Facebook
Twitter
LinkedIn
YouTube
Other

---