

Security {Reviewer}



Security Reviewer Suite

***Security Reviewer
Dynamic Reviewer
Firmware Reviewer
Team Reviewer
Effort Estimation
Mobile***

Multi-Language Code Analysis



Security Reviewer Suite

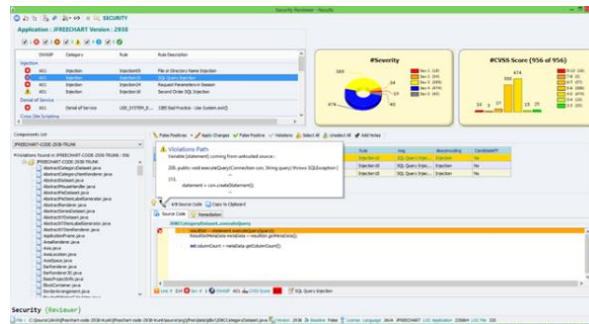
Security Reviewer Products

- Security Reviewer
- Quality Reviewer
- Mobile Reviewer
- Firmware Reviewer
- Team Reviewer

Features

- Classified Security Vulnerabilities
- Dynamic Exploit Analysis
- Correlation b/w Static and Dynamic Analysis
- Dead Code detection
- False Positives optimization
- **SQLALE Quality Model**
- **Effort Estimation**

WMFP, OMG Automated FP, COCOMO, Revic, COSMIC FFP



Security Reviewer is a Static and Dynamic Analysis toolset for all kind of apps (Mobile Android and iOS included) to support quality and secure code inspection, including software Metrics measurement, SQALE-based quality management solutions, Effort Estimation and Firmware Analysis

Features

- **SAST** (Static Analysis) Works on un-compiled source code, with 40+ programming languages support, and catches problems that other suites miss. More importantly, provides defects detection earlier in the development cycle, when they are easier and less expensive to fix, with almost **Zero False Positives**.
- **Quality metrics** measurement and Application Portfolio Evaluation (100+ metrics like McCabe® complexity, Size and structure Metrics, Halstead Metrics, SEI maintainability, CK/Mood Metrics, Effort Estimation)
- **Attack Vectors** and Exploit scripts are automatically generated during Static Analysis. The included Dynamic Exploit Analysis provides a surface attack, acting like a smart sandbox, using Dynamic Syntax Tree
- **Software Composition Analysis** (SCA) provides a comprehensive management and reporting about frameworks and libraries dependencies, both by legal and security point-of-view
- **DAST** (Dynamic Analysis) Distributed Lightweight Pen-Test directly from you Browser
- **Firmware Analysis** from a Firmware image detects the vulnerabilities and possible exploit. No need of physical device
- **Vulnerability Management** & Tracking (SAST and DAST integration) with Team Reviewer. Support of 20+ third party tools
- Enhanced **SQLALE** integrated defect analysis reports (1000+ effective security and dead code rules). What-if Analysis. Risk Index. **Technical Debt**. Quality Index. Confidence Factor. See: www.sqlale.org/tools



Description	View	SQLALE	LOC	%	More Metrics	Changeable	Partially	Not Changeable	Security	Level	%
Technical Debt	45.56%	16	10449	90	16	38.28	31.37	30.35	High	100	100%
Code of Code	44.62%	Method	0	0	0	0	0	0	Low	0	0%
File Size	44%	Method	0	0	0	0	0	0	Low	0	0%
File Size	44%	Method	0	0	0	0	0	0	Low	0	0%

Compliance

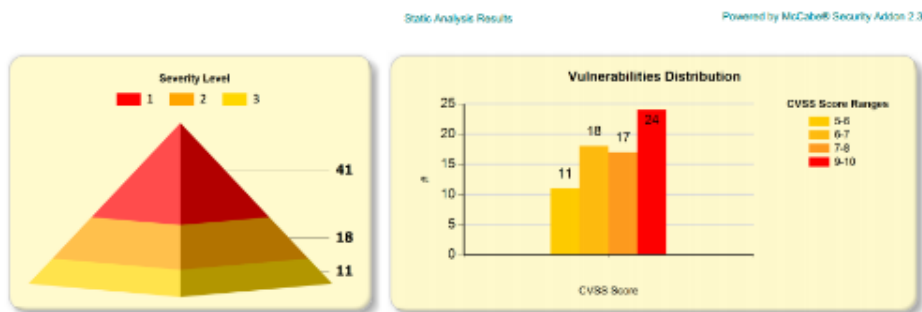
Compliance Reporting

- OWASP, CWE, PCI-DSS, WASC, MISRA
- CVE and CVSS
- Vulnerable Libraries detection
- Template- based Reporting
- SQALE enhanced Reporting
- Outsourcer Rating
- Confidence Factor

Languages

- C/C++
- JAVA, JSP, JSF
- JavaScript, TypeScript
- C#, Vb.Net
- PHP
- Python
- R, Rust, Clojure
- Ruby, Groovy
- Scala, GO, Kotlin
- ABAP, COBOL
- Flex, ActionScript
- XML, XPath
- Mobile Android
- Apple Objective-C SWIFT
- sh, PowerShell, Auto-IT, LUA

Each vulnerability detected, is classified using OWASP Top 10 2021, Mobile OWASP Top 10 2016, PCI-DSS 3.2.1, WASC, MISRA, CVSS v3.1, CVE and CWE compliance standards:



Security

The objective of this Security Audit was to identify different types of potentially exploitable vulnerabilities in the source code. To perform the audit we relied on the expertise of our auditors and we based the analysis on the following test plan:

- Data flow analysis: identifies the input parameters of the applications (which can be manipulated by an attacker) and detects the absence of validation at any point in the application in question.
- Semantic analysis: validates the use of potentially dangerous code itself; that is, calls to vulnerable functions and procedures.
- Structural analysis: detects problems due to incorrect decisions in the organization of code.
- Fingerprint analysis: detects all vulnerable libraries and frameworks used by the application.

These checks were first performed by automated procedures, then manual intervention was needed, taking into account the established risk on the application.

Platform and System Requirements

- ✚ Windows 10 or higher. Windows 2008 SR2 or higher, Linux, MacOS.
- ✚ .NET Core 5
- ✚ At least 4GB RAM or minimum RAM required by Host System
- ✚ 1GB of free C: disk space during running

Licensing and IDE

- ✚ Server License. Concurrent users with RESTful API Server. (Yearly subscription)
- ✚ Desktop license (Yearly subscription)
- ✚ DevOps (Jenkins, GitLab, Azure DevOps, Amazon AWS, Ant, Maven, Gradle, etc.)
- ✚ Firmware Reviewer (Cloud subscription)
- ✚ Effort Estimation add-in, Software Composition Analysis (Yearly subscription)
- ✚ Enterprise License. Includes Eclipse, IBM Rational Team Concert, MS Visual Studio

www.securityreviewer.net

Via della Pace, 154
58100 Grosseto
Italy

info@securityreviewer.com

Visit our Knowledge Center:
securityreviewer.atlassian.net

Main capability of our technology is to afford developers the ability to scrub their code of obvious and not-so-obvious weaknesses as they work, before they submit their code for check-in and more formal down-stream validation procedures.

Security Reviewer is an Italian startup company beginning its path on app security in 2001. Security Reviewer is made of a group of senior professionals with strong Application Security skills, some of them formerly University Professors worked on applying classic security methodologies (OWASP, OSSTM, CVSS notation) on Web apps and Mobile environments.