

mre/awesome-static-analysis

[Originalartikel](#)

[Backup](#)

```
<html> <div class=„Box-body p-6“> <article class=„markdown-body entry-content“
itemprop=„text“><p><a target=„_blank“ rel=„noopener noreferrer“
href=„https://github.com/mre/awesome-static-analysis/blob/master/awesome.png“><img
src=„https://github.com/mre/awesome-static-analysis/raw/master/awesome.png“ alt=„Logo“
class=„cl“/></a></p> <blockquote> <p>Static program analysis is the analysis of computer
software that is performed without actually executing programs &#8212; <a
href=„https://en.wikipedia.org/wiki/Static_program_analysis“ rel=„nofollow“>Wikipedia</a></p>
</blockquote> <p>This is a collection of static analysis tools and code quality checkers. Pull requests
are very welcome!<br/><strong>Note: &#169;&#65039; stands for proprietary software. All other
tools are Open Source.</strong><br/>Also check out the sister project, <a
href=„https://github.com/mre/awesome-dynamic-analysis“>awesome-dynamic-analysis</a>.</p>
<ul><li><a href=„https://github.com/mre/awesome-static-analysis#programming-
languages“>Programming Languages</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#multiple-languages“>Multiple
languages</a></li> <li><a href=„https://github.com/mre/awesome-static-
analysis#other“>Other</a> <ul><li><a
href=„https://github.com/mre/awesome-static-analysis#build-tools“>Build tools</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#binaries“>Binaries</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#containers“>Containers</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#config-files“>Config Files</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#configuration-management“>Configuration
Management</a></li> <li><a href=„https://github.com/mre/awesome-static-
analysis#css“>CSS</a></li> <li><a href=„https://github.com/mre/awesome-static-
analysis#gherkin“>Gherkin</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#html“>HTML</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#ide-plugins“>IDE Plugins</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#latex“>LaTeX</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#makefiles“>Makefiles</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#markdown“>Markdown</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#mobile“>Mobile</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#packages“>Packages</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#supporting-tools“>Supporting
Tools</a></li> <li><a href=„https://github.com/mre/awesome-static-analysis#template-
languages“>Template Languages</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#translation“>Translation</a></li> <li><a
href=„https://github.com/mre/awesome-static-analysis#web-services“>Web services</a></li>
<li><a href=„https://github.com/mre/awesome-static-analysis#writing“>Writing</a></li>
</ul></li> <li><a href=„https://github.com/mre/awesome-static-analysis#more-collections“>More
Collections</a></li> </ul> <h2><a id=„user-content-ada“ class=„anchor“ aria-hidden=„true“
href=„https://github.com/mre/awesome-static-analysis#ada“><svg class=„octicon octicon-link“
viewbox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/></a>Ada</h2>
<ul><li><a href=„http://www.adacore.com/codepeer“ rel=„nofollow“>Codepeer</a> - detects run-
time and logic errors</li> <li><a href=„https://www.mathworks.com/products/polyspace-ada.html“
rel=„nofollow“>Polyspace for Ada</a> &#169;&#65039; - provide code verification that proves the
```

absence of overflow, divide-by-zero, out-of-bounds array access, and certain other run-time errors in source code.

- <http://www.spark-2014.org/about> - Static analysis and formal verification toolset for Ada
- <https://scitools.com/ada-programming-essential/> - IDE that provides code analysis, standards testing, metrics, graphing, dependency analysis and more for Ada and VHDL.

<https://github.com/mre/awesome-static-analysis#awk>



Awk

- https://www.gnu.org/software/gawk/manual/html_node/Options.html - warns about constructs that are dubious or nonportable to other awk implementations.

<https://github.com/mre/awesome-static-analysis#cc>



C/C++

- <http://www.cprover.org/cbmc/> - bounded model-checker for C programs, user-defined assertions, standard assertions, several coverage metric analyses
- <http://clang.llvm.org/extra/clang-tidy/> - clang static analyser
- <https://github.com/MetricsGrimoire/CMetrics> - Measures size and complexity for C files
- <https://www.grammotech.com/products/codesonar> - Advanced, whole program, deep path, static analysis of C and C++ with easy-to-understand explanations and code and path visualization.
- <https://github.com/jameysharp/corrode> - Semi-automatic translation from C to Rust. Could reveal bugs in the original implementation by showing Rust compiler warnings and errors.
- <https://github.com/danmar/cppcheck> - static analysis of C/C++ code
- <https://www.cppdepend.com> - Measure, query and visualize your code and avoid unexpected issues, technical debt and complexity.
- <https://github.com/google/styleguide/tree/gh-pages/cpplint> - automated C++ checker that follows Google's style guide
- <https://github.com/dspinellis/cqmetrics> - quality metrics for C code
- <https://www.spinellis.gr/cscout/> - complexity and quality metrics for C and C preprocessor code
- <http://www.dwheeler.com/flipfinder/> - finds possible security weaknesses
- <http://l2program.co.uk/category/flint> - cross-platform, zero-dependency port of flint, a lint program for C++ developed and used at Facebook.
- <http://frama-c.com/> - a sound and extensible static analyzer for C code
- <https://github.com/nasa-sw-vnv/ikos> - a sound static analyzer for C/C++ code based on LLVM
- <http://oclint.org/> - static analysis of C/C++ code
- <https://www.mathworks.com/products/polyspace-bug-finder.html> - identifies run-time errors, concurrency issues, security vulnerabilities, and other defects in C and C++ embedded software.
- <https://www.mathworks.com/products/polyspace-code-prover.html> - provide code verification that proves the absence of overflow, divide-by-zero, out-of-bounds array access, and certain other run-time errors in C and C++ source code.
- <https://clang-analyzer.llvm.org/scan-build.html> - Analyzes C/C++ code using LLVM at compile-time
- <https://github.com/ravenexp/splint> - Annotation-assisted static program checker
- <https://bitbucket.org/verateam/vera/wiki/Introduction> - Vera++ is a programmable tool for verification, analysis and

transformation of C++ source code.

- <https://github.com/mre/awesome-static-analysis#c>

- <https://github.com/DotNetAnalyzers> - .NET Analyzers - An organization for the development of analyzers (diagnostics and code fixes) using the .NET Compiler Platform.
- <https://carc.codeplex.com/> - Code Analysis Rule Collection - Contains a set of diagnostics, code fixes and refactorings built on the Microsoft .NET Compiler Platform „Roslyn“.
- <https://github.com/code-cracker/code-cracker> - code-cracker - An analyzer library for C# and VB that uses Roslyn to produce refactorings, code analysis, and other niceties.
- <https://www.devexpress.com/products/coderush/> - CodeRush - Code creation, debugging, navigation, refactoring, analysis and visualization tools that use the Roslyn engine in Visual Studio 2015 and up.
- <https://github.com/DustinCampbell/CSharpEssentials> - CSharpEssentials - C# Essentials is a collection of Roslyn diagnostic analyzers, code fixes and refactorings that make it easy to work with C# 6 language features.
- <http://www.designite-tools.com> - Designite - Designite is a software design quality assessment tool. It supports detection of implementation and design smells, computation of various code quality metrics, and trend analysis.
- <http://www.mono-project.com/docs/tools+libraries/tools/gendarme/> - Gendarme - Gendarme inspects programs and libraries that contain code in ECMA CIL format (Mono and .NET).
- <http://www.ndepend.com/> - NDepend - Measure, query and visualize your code and avoid unexpected issues, technical debt and complexity.
- <https://github.com/pumasecurity/puma-scan> - Puma Scan - Puma Scan provides real time secure code analysis for common vulnerabilities (XSS, SQLi, CSRF, LDAPi, crypto, deserialization, etc.) as development teams write code in Visual Studio.
- <http://vsrefactoringessentials.com/> - Refactoring Essentials - The free Visual Studio 2015 extension for C# and VB.NET refactorings, including code best practice analyzers.
- <https://www.jetbrains.com/resharper/> - ReSharper - Extends Visual Studio with on-the-fly code inspections for C#, VB.NET, ASP.NET, JavaScript, TypeScript and other technologies.
- <https://dotnet-security-guard.github.io/> - Roslyn Security Guard - Project that focus on the identification of potential vulnerabilities such as SQL injection, cross-site scripting (XSS), CSRF, cryptography weaknesses, hardcoded passwords and many more.
- <https://github.com/JosefPihrt/Roslynator/> - Roslynator - A collection of 190+ analyzers and 190+ refactorings for C#, powered by Roslyn.
- <https://security-code-scan.github.io/> - Security Code Scan - Security code analyzer for C# and VB.NET. Detects various security vulnerability patterns: SQLi, XSS, CSRF, XXE, Open Redirect, etc.
- <https://vs.sonarlint.org/> - SonarLint for Visual Studio - SonarLint is an extension for Visual Studio 2015 and 2017 that provides on-the-fly feedback to developers on new bugs and quality issues injected into .NET code.
- <https://github.com/Vannevelj/VSDiagnostics> - VSDiagnostics - A collection of static analyzers based on Roslyn that integrate with VS.
- <https://github.com/Wintellect/Wintellect.Analyzers> - Wintellect.Analyzers - .NET Compiler Platform („Roslyn“) diagnostic analyzers and code fixes.

<https://github.com/mre/awesome-static-analysis#crystal>


Crystal

- <https://github.com/veelenga/ameba> - ameba - A static code analysis tool for Crystal

<https://crystal-lang.org/> - The Crystal compiler has built-in linting functionality.

<https://github.com/mre/awesome-static-analysis#dlang>  Dlang

<https://github.com/dlang-community/D-Scanner> - D-scanner

<https://github.com/mre/awesome-static-analysis#elixir>  Elixir

<https://github.com/rrrene/credo> - A static code analysis tool with a focus on code consistency and teaching.

<https://github.com/nccgroup/sobelow> - Security-focused static analysis for the Phoenix Framework

<https://github.com/mre/awesome-static-analysis#erlang>  Erlang

<https://github.com/inaka/elvis> - Erlang Style Reviewer

<https://github.com/mre/awesome-static-analysis#f>  F#

<https://github.com/mre/awesome-static-analysis#fortran>  Fortran

<https://github.com/lequal/i-CodeCNES> - i-Code CNES for Fortran

<https://github.com/mre/awesome-static-analysis#go>  Go

<https://github.com/tsenart/deadcode> - deadcode

<https://github.com/nickng/dingo-hunter> - dingo-hunter

<https://github.com/mibk/dupl> - dupl

<https://github.com/kisielk/errcheck> - errcheck

<https://github.com/lafolle/flen> - flen

<https://github.com/GoASTScanner/gas> - gas

<https://github.com/alecthomas/gometalinter> - Go Meta Linter

https://golang.org/cmd/vet/#hdr-Shadowed_variables - go tool vet -shadow

<https://golang.org/cmd/vet/> - go vet

<https://github.com/go-critic/go-critic> - go-critic

<https://github.com/dominikh/go-tools/tree/master/cmd/staticcheck> - go-staticcheck

<https://golang.org/pkg/go/ast/> - go/ast

<https://github.com/jgautheron/goconst> - goconst

<https://github.com/fzipp/gocyclo> - gocyclo

<https://golang.org/cmd/gofmt/> - gofmt -s

<https://godoc.org/golang.org/x/tools/cmd/goimports> - goimports

missing or unreferenced package imports.

- <https://github.com/golang/lint> - golint - Prints out coding style mistakes in Go source code.
- <https://github.com/wgliang/goreporter> - goreporter - concurrently runs many linters and normalises their output to a report.
- <https://github.com/linuxerwang/goroutine-inspect> - goroutine-inspect - An interactive tool to analyze Golang goroutine dump.
- <https://github.com/dominikh/go-tools/tree/master/cmd/gosimple> - gosimple - Report simplifications in code.
- <https://golang.org/x/tools/cmd/gotype> - gotype - Syntactic and semantic analysis similar to the Go compiler.
- <https://github.com/gordonklaus/ineffassign> - ineffassign - Detect ineffectual assignments in Go code
- <https://github.com/mvdan/interfacer> - interfacer - Suggest narrower interfaces that can be used.
- <https://github.com/walle/lll> - lll - Report long lines.
- <https://github.com/mdempsky/maligned> - maligned - Detect structs that would take less memory if their fields were sorted.
- <https://github.com/dominikh/go-tools/tree/master/cmd/megacheck> - megacheck - Run staticcheck, gosimple and unused, sharing work.
- <https://github.com/client9/misspell> - misspell - Finds commonly misspelled English words.
- <https://github.com/alexkohler/nakedret> - nakedret - Finds naked returns.
- <https://github.com/alexkohler/prealloc> - prealloc - Finds slice declarations that could potentially be preallocated.
- <https://github.com/mgechev/revive> - revive - Fast, configurable, extensible, flexible, and beautiful linter for Go. Drop-in replacement of golint.
- <https://github.com/stripe/safesql> - safesql - Static analysis tool for Golang that protects against SQL injections.
- <https://github.com/opennota/check> - structcheck - Find unused struct fields.
- <http://golang.org/pkg/testing/> - test - Show location of test failures from the stdlib testing module.
- <https://github.com/stretchr/testify> - testify - Show location of failed testify assertions.
- <https://github.com/mdempsky/unconvert> - unconvert - Detect redundant type conversions.
- <https://github.com/alexkohler/unimport> - unimport - Finds unnecessary import aliases
- <https://github.com/mvdan/unparam> - unparam - Find unused function parameters.
- <https://github.com/dominikh/go-tools/tree/master/cmd/unused> - unused - Find unused variables.
- <https://github.com/opennota/check> - varcheck - Find unused global variables and constants.

[**Groovy**](https://github.com/mre/awesome-static-analysis#groovy)

- <https://github.com/CodeNarc/CodeNarc> - CodeNarc - a static analysis tool for Groovy source code, enabling monitoring and enforcement of many coding standards and best practices

[**Haskell**](https://github.com/mre/awesome-static-analysis#haskell)

- <https://github.com/ndmitchell/hlint> - HLint - HLint is a tool for suggesting possible improvements to Haskell code.

[**Haxe**](https://github.com/mre/awesome-static-analysis#haxe)

- <https://github.com/HaxeCheckstyle/haxe-checkstyle> - Haxe Checkstyle - A static analysis

tool to help developers write Haxe code that adheres to a coding standard.

[id=„user-content-java“ class=„anchor“ aria-hidden=„true“ href=„https://github.com/mre/awesome-static-analysis#java“](https://github.com/mre/awesome-static-analysis#java) <svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Java

- <https://www.archunit.org/> rel=„nofollow“>ArchUnit - Unit test your Java architecture
- <https://github.com/typetools/checker-framework/>>Checker Framework - Pluggable type-checking for Java <http://checkerframework.org/> rel=„nofollow“>http://checkerframework.org/- <https://github.com/checkstyle/checkstyle>>checkstyle - checking Java source code for adherence to a Code Standard or set of validation rules (best practices)
- <http://www.spinellis.gr/sw/ckjm/> rel=„nofollow“>ckjm - calculates Chidamber and Kemerer object-oriented metrics by processing the bytecode of compiled Java files
- <https://github.com/classgraph/classgraph>>ClassGraph - a classpath and module path scanner for querying or visualizing class metadata or class relatedness
- <https://github.com/google/error-prone>>Error-prone - Catch common Java mistakes as compile-time errors
- <https://github.com/mebigfatguy/fb-contrib>>fb-contrib - A plugin for FindBugs with additional bug detectors
- <https://find-security.github.io/> rel=„nofollow“>Find Security Bugs - IDE/SonarQube plugin for security audits of Java web applications.
- <https://github.com/cuplv/hopper>>Hopper - A static analysis tool written in scala for languages that run on JVM
- <https://github.com/amaembo/huntbugs>>HuntBugs - Bytecode static analyzer tool based on Procyon Compiler Tools aimed to supersede FindBugs.
- <https://www.jarchitect.com> rel=„nofollow“>JArchitect ©️ - Measure, query and visualize your code and avoid unexpected issues, technical debt and complexity.
- <http://www.cprover.org/jbmc/> rel=„nofollow“>JPMC - bounded model-checker for Java (bytecode), verifies user-defined assertions, standard assertions, several coverage metric analyses
- <https://github.com/uber/NullAway>>NullAway - Type-based null-pointer checker with low build-time overhead; an <http://errorprone.info/> rel=„nofollow“>Error Prone plugin
- https://www.owasp.org/index.php/OWASP_Dependency_Check rel=„nofollow“>OWASP Dependency Check - Checks dependencies for known, publicly disclosed, vulnerabilities.
- <https://sable.github.io/soot/> rel=„nofollow“>Soot - A framework for analyzing and transforming Java and Android applications.
- <https://github.com/INRIA/spoon>>Spoon - Library to write your own static analyses and architectural rule checkers for Java. Can be integrated in Maven and Gradle.
- <https://spotbugs.github.io/> rel=„nofollow“>SpotBugs - SpotBugs is FindBugs' successor. A tool for static analysis to look for bugs in Java code.

[id=„user-content-javascript“ class=„anchor“ aria-hidden=„true“ href=„https://github.com/mre/awesome-static-analysis#javascript“](https://github.com/mre/awesome-static-analysis#javascript) <svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>JavaScript

- <https://github.com/codecombat/aether>>aether - Lint, analyze, normalize, transform, sandbox, run, step through, and visualize user JavaScript, in node or the browser.
- <https://github.com/google/closure-linter>>ClosureLinter - ensures that all of your project's JavaScript code follows the guidelines in the Google JavaScript Style Guide. It can also automatically fix many common errors
- <https://github.com/clutchski/coffeelint>>coffeelint - A style checker that helps keep CoffeeScript code clean and consistent.
- <https://github.com/jared-stilwell/complexity-report>>complexity-report - Software complexity analysis for JavaScript projects
- <https://deepscan.io> rel=„nofollow“>DeepScan ©️ - An analyzer for JavaScript which targets runtime

errors and quality issues rather than coding conventions.

- <https://github.com/jared-stilwell/escomplex> - Software complexity analysis of JavaScript-family abstract syntax trees.
- <https://github.com/eslint/eslint> - A fully pluggable tool for identifying and reporting on patterns in JavaScript
- <https://github.com/jquery/esprima> - ECMAScript parsing infrastructure for multipurpose analysis
- <https://flow.org/> - A static type checker for JavaScript.
- <https://github.com/jshint/jshint> - detect errors and potential problems in JavaScript code and enforce your team's coding conventions
- <https://github.com/douglascrockford/JSLint> - The JavaScript Code Quality Tool
- <https://github.com/dpnishant/jsprime> - static security analysis tool
- <https://github.com/ajinabraham/NodeJSScan> - NodeJSScan is a static security code scanner for Node.js applications.
- <https://github.com/es-analysis/plato> - Visualize JavaScript source complexity
- <https://github.com/prettier/prettier> - An opinionated code formatter.
- <https://github.com/jden/quality> - zero configuration code and module linting
- <https://github.com/RetireJS/retire.js> - Scanner detecting the use of JavaScript libraries with known vulnerabilities
- <http://standardjs.com/> - An npm module that checks for Javascript Styleguide issues
- <https://github.com/sindresorhus/xo> - Enforce strict code style. Never discuss code style on a pull request again!
- <https://github.com/calmh/yardstick> - Javascript code metrics

[Kotlin](https://github.com/mre/awesome-static-analysis#kotlin)

- <https://github.com/arturbosch/detekt> - Static code analysis for Kotlin code.
- <https://github.com/shyiko/ktlint> - An anti-bikeshedding Kotlin linter with built-in formatter

[Lua](https://github.com/mre/awesome-static-analysis#lua)

- <https://github.com/mpeterv/luacheck> - A tool for linting and static analysis of Lua code.

[MATLAB](https://github.com/mre/awesome-static-analysis#matlab)

- <https://de.mathworks.com/help/matlab/ref/mlint.html> - Check MATLAB code files for possible problems.

[Perl](https://github.com/mre/awesome-static-analysis#perl)

- <http://search.cpan.org/%7Eethanjef/Perl-Critic-1.126/lib/Perl/Critic.pm> - Critique Perl source code for best-practices.

[PHP](https://github.com/mre/awesome-static-analysis#php)

- <https://github.com/mihaeu/dephend> - Dependency analysis tool
- <https://github.com/sensiolabs-de/deprecation-detector> - Finds usages of deprecated (Symfony) code
- <https://github.com/sensiolabs-de/deptract> - Enforce rules for dependencies

between software layers.

- <https://github.com/Halleck45/DesignPatternDetector> - DesignPatternDetector - detection of design patterns in PHP code
- <https://github.com/Symplify/EasyCodingStandard> - EasyCodingStandard - combine https://github.com/squizlabs/PHP_CodeSniffer and <https://github.com/FriendsOfPHP/PHP-CS-Fixer>
- <https://github.com/exakat/exakat> - exakat - An automated code reviewing engine for PHP
- <https://github.com/phpro/grumphp> - GrumPHP - checks code on every commit
- <https://github.com/Trismegiste/Mondrian> - Mondrian - a set of static analysis and refactoring tools which use graph theory
- <https://github.com/JakubOnderka/PHP-Parallel-Lint> - parallel-lint - This tool checks syntax of PHP files faster than serial check with a fancier output.
- <https://github.com/psecio/parse> - Parse - A Static Security Scanner
- <https://pdepend.org/> - pdepend - Calculates software metrics like cyclomatic complexity for PHP code.
- <https://github.com/etsy/phan> - phan - a modern static analyzer from etsy
- <https://github.com/rskuipers/php-assumptions> - PHP Assumptions - Checks for weak assumptions
- <http://cs.sensiolabs.org/> - PHP Coding Standards Fixer - Fixes your code according to standards like PSR-1, PSR-2, and the Symfony standard.
- <https://github.com/kalessil/phpinspectionsea> - Php Inspections (EA Extended) - A Static Code Analyzer for PHP.
- <https://github.com/QafooLabs/php-refactoring-browser> - PHP Refactoring Browser - Refactoring helper
- <https://github.com/tomzx/php-semver-checker> - PHP Semantic Versioning Checker - Suggests a next version according to semantic versioning
- <https://github.com/nikic/PHP-Parser> - PHP-Parser - A PHP parser written in PHP
- <https://github.com/Andrewsville/PHP-Token-Reflection> - PHP-Token-Reflection - Library emulating the PHP internal reflection
- <https://github.com/sstalle/php7cc> - php7cc - PHP 7 Compatibility Checker
- <https://github.com/Alexia/php7mar> - php7mar - assist developers in porting their code quickly to PHP 7
- https://github.com/squizlabs/PHP_CodeSniffer - PHP_CodeSniffer - detects violations of a defined set of coding standards
- <https://github.com/wapmorgan/PhpCodeAnalyzer> - phpca - Finds usage of non-built-in extensions
- <http://wapmorgan.github.io/PhpCodeFixer/> - phpcf - Finds usage of deprecated PHP features
- <https://github.com/sebastianbergmann/phpcpd> - phpcpd - Copy/Paste Detector for PHP code.
- <https://github.com/sebastianbergmann/phpdcd> - phpdc - Dead Code Detector (DCD) for PHP code.
- <https://github.com/mamuz/PhpDependencyAnalysis> - PhpDependencyAnalysis - builds a dependency graph for a project
- <https://github.com/dunglas/phpdoc-to-typehint> - phpdoc-to-typehint - Add scalar type hints and return types to existing PHP projects using PHPDoc annotations
- <https://www.phpdoc.org/> - phpDocumentor - Analyzes PHP source code to generate documentation
- <https://phpmd.org/> - PHPMD - finds possible bugs in your code
- <http://www.phpmetrics.org/> - PhpMetrics - Calculates and visualizes various code quality metrics
- <https://github.com/povils/phpmnd> - phpmnd - Helps to detect magic numbers
- <https://github.com/EdgedesignCZ/phpqa> - PHPQA - A tool for running QA tools (phploc, phpcpd, phpcs, pdepend, phpmd, phpmetrics)
- <https://github.com/jakzal/phpqa> - phpqa - jakzal - Many tools for PHP static analysis in

one container

- <https://github.com/jmolivas/phpqa> - PHPQA all-in-one Analyzer CLI tool
- <https://github.com/ovr/phpsa> - Static analysis tool for PHP.
- <https://github.com/phpstan/phpstan> - PHP Static Analysis Tool - discover bugs in your code without running it!
- <https://github.com/designsecurity/progpilot> - Progpilot - A static analysis tool for security purposes
- <https://getpsalm.org/> - Psalm - Static analysis tool for finding type errors in PHP applications
- <https://github.com/Qafoo/QualityAnalyzer> - Qafoo Quality Analyzer - Visualizes metrics and source code
- <https://github.com/ripsscanner/rips> - RIPS - A static source code analyser for vulnerabilities in PHP scripts
- <https://github.com/ircmaxell/Tuli> - Tuli - A static analysis engine
- <https://github.com/asm89/twig-lint> - twig-lint is a lint tool for your twig files.
- https://www.owasp.org/index.php/OWASP_WAP-Web_Application_Protection - WAP - Tool to detect and correct input validation vulnerabilities in PHP (4.0 or higher) web applications and predicts false positives by combining static analysis and data mining.

Python

- <https://github.com/mre/awesome-static-analysis#python>
- <https://github.com/PyCQA/bandit> - bandit - a tool to find common security issues in Python code
- <https://github.com/ambv/black> - Black - The uncompromising Python code formatter
- <https://github.com/davidhalter/jedi> - jedi - autocompletion/static analysis library for Python
- https://github.com/lyft/linty_fresh - linty fresh - parse lint errors and report them to Github as comments on a pull request
- <https://github.com/PyCQA/mccabe> - mccabe - check McCabe complexity
- <https://github.com/python/mypy> - mypy - a static type checker that aims to combine the benefits of duck typing and static typing, frequently used with
- <https://github.com/Instagram/MonkeyType> - MonkeyType
- <https://github.com/uber/py-find-injection> - py-find-injection - find SQL injection vulnerabilities in Python code
- <https://github.com/PyCQA/pycodestyle> - pycodestyle - (formerly

pep8

) check Python code against some of the style conventions in PEP 8

- <https://github.com/PyCQA/pydocstyle> - pydocstyle - check compliance with Python docstring conventions
- <https://github.com/pyflakes/pyflakes> - pyflakes - check Python source files for errors
- <https://github.com/PyCQA/pylint> - pylint - looks for programming errors, helps enforcing a coding standard and sniffs for some code smells. It additionally includes

pyreverse

(an UML diagram generator) and

symilar

(a similarities checker).

- <https://github.com/facebook/pyre-check> - pyre-check - A fast, scalable type checker for large Python codebases
- <https://github.com/regebro/pyroma> - pyroma - rate how well a Python project complies

with the best practices of the Python packaging ecosystem, and list issues that could be improved

- [PyT](https://github.com/python-security/pyt) - Python Taint - A static analysis tool for detecting security vulnerabilities in Python web applications.
- [radon](https://github.com/rubik/radon) - a Python tool that computes various metrics from the source code
- [vulture](https://github.com/jendrikseipp/vulture) - find unused classes, functions and variables in Python code
- [xenon](https://github.com/rubik/xenon) - monitor code complexity using
- [radon](https://github.com/rubik/radon)

radon

[Python wrappers](https://github.com/mre/awesome-static-analysis#python-wrappers)

- [ciocheck](https://github.com/ContinuumIO/ciocheck) - linter, formatter and test suite helper. As a linter, it is a wrapper around

pep8

,

pydocstyle

,

flake8

, and

pylint

.

- [flake8](https://github.com/PyCQA/flake8) - a wrapper around

pyflakes

,

pycodestyle

and

mccabe

- [prospector](https://github.com/landscapeio/prospector) - a wrapper around

pylint

,

pep8

,

mccabe

and others

- <https://github.com/mre/awesome-static-analysis#r> - Static Code Analysis for R
- <https://github.com/jimhester/lintr> - Static Code Analysis for R
- <https://github.com/mre/awesome-static-analysis#rpg> - Static Code Analysis for RPG III and RPG IV versions (including free-form)
- <https://www.sourcemeter.com/resources/rpg/> - SourceMeter
- <https://github.com/mre/awesome-static-analysis#ruby> - Static analysis security vulnerability scanner for Ruby on Rails applications
- <https://github.com/square/cane> - Code quality threshold checking as part of your build
- <https://github.com/thesp0nge/dawnscanner> - a static analysis security scanner for ruby written web applications. It supports Sinatra, Padrino and Ruby on Rails frameworks.
- <https://github.com/seattlerb/flay> - Flay analyzes code for structural similarities.
- <https://github.com/seattlerb/flog> - Flog reports the most tortured code in an easy to read pain report. The higher the score, the more pain the code is in.
- <https://github.com/michaelledgar/laser> - Static analysis and style linter for Ruby code.
- <https://github.com/codegram/pelusa> - Static analysis Lint-type tool to improve your OO Ruby code
- <https://github.com/apiology/quality> - Runs quality checks on your code using community tools, and makes sure your numbers don't get any worse over time.
- <https://github.com/soutaro/querly> - Pattern Based Checking Tool for Ruby
- <https://github.com/troessner/reek> - Code smell detector for Ruby
- <https://github.com/rubocop-hq/rubocop> - A Ruby static code analyzer, based on the community Ruby style guide.
- <https://github.com/blazeeboy/rubrowser> - Ruby classes interactive dependency graph generator.
- <https://github.com/YorickPeterse/ruby-lint> - Static code analysis for Ruby
- <https://github.com/whitesmith/rubycritic> - A Ruby code quality reporter
- https://github.com/makaroni4/sandi_meter - Static analysis tool for checking Ruby code for Sandi Metz' rules.

Rust

- <https://github.com/Manishearth/rust-clippy> - a code linter to catch common mistakes and improve your Rust code
- <https://github.com/Kha/electrolysis> - A tool for formally verifying Rust programs by transpiling them into definitions in the Lean theorem prover.
- <https://github.com/mcarton/rust-herbie-lint> - Adds warnings or errors to your crate when using a numerically unstable floating point expression.
- <https://github.com/AtomLinter/linter-rust> - Linting your Rust-files in Atom, using rustc and cargo
- <https://github.com/rust-lang-nursery/rls> - Rust Language

Server - Supports functionality such as 'goto definition', symbol search, reformatting, and code completion, and enables renaming and refactorings.

<https://github.com/killercup/rustfix> - rustfix - read and apply the suggestions made by rustc (and third-party lints, like those offered by clippy).

<https://github.com/mre/awesome-static-analysis#scala> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> Scala

<https://github.com/HairyFotr/linter> - linter - Linter is a Scala static analysis compiler plugin which adds compile-time checks for various possible bugs, inefficiencies, and style problems.

<http://www.scalastyle.org> - Scalastyle - Scalastyle examines your Scala code and indicates potential problems with it.

<https://github.com/sksamuel/scapegoat> - scapegoat - Scala compiler plugin for static code analysis

<https://github.com/puffnfresh/wartremover> - WartRemover - a flexible Scala code linting tool.

<https://github.com/mre/awesome-static-analysis#shell> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> Shell

<https://github.com/lequal/i-CodeCNES> - i-Code CNES for Shell - An open source static code analysis tool for Shell and Fortran (77 and 90).

<https://github.com/koalaman/shellcheck> - shellcheck - ShellCheck, a static analysis tool that gives warnings and suggestions for bash/sh shell scripts

<https://github.com/mre/awesome-static-analysis#solidity> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> Solidity

<https://github.com/duaraghav8/Solium> - solium - Solium is a linter to identify and fix style and security issues in Solidity smart contracts

<https://github.com/mre/awesome-static-analysis#sql> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> SQL

<https://github.com/jarulraj/sqlcheck> - sqlcheck - Automatically identify anti-patterns in SQL queries

<https://github.com/purcell/sqlint> - sqlint - Simple SQL linter

<https://github.com/tsqlint/tsqlint> - T-SQL-specific linter

<https://github.com/ashleyglee/TSqlRules> - TSqlRules - TSQL Static Code Analysis Rules for SQL Server

<https://github.com/mre/awesome-static-analysis#swift> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> Swift

<https://github.com/nicklockwood/SwiftFormat> - SwiftFormat - A library and command-line formatting tool for reformatting Swift code

<https://github.com/realm/SwiftLint> - SwiftLint - A tool to enforce Swift style and conventions

<https://github.com/sleekbyte/tailor> - Tailor - A static analysis and lint tool for source code written in Apple's Swift programming language.

<https://github.com/mre/awesome-static-analysis#typescript> <svg class="octicon octicon-link" viewBox="0 0 16 16" version="1.1" width="16" height="16" aria-hidden="true"/> TypeScript

<https://github.com/mgechev/codelyzer> - Codelyzer - A set of tslint rules for static code analysis of Angular 2 TypeScript projects.

<https://github.com/palantir/tslint> - TSLint - An extensible linter for the TypeScript language.

<https://github.com/Glavin001/tslint-clean-code> - tslint-clean-code - A set of TSLint rules inspired by the Clean Code handbook.

href=„<https://github.com/Microsoft/tslint-microsoft-contrib>“>tslint-microsoft-contrib - A set of tslint rules for static code analysis of TypeScript projects maintained by Microsoft. <h2><a id=„user-content-vbscript“ class=„anchor“ aria-hidden=„true“ href=„<https://github.com/mre/awesome-static-analysis#vbscript>“><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>VBScript</h2> <a href=„<http://patterson-consulting.net/tds>“ rel=„nofollow“>Test Design Studio ©️ - A full IDE with static code analysis for Micro Focus Unified Functional Testing VBScript-based automated tests. <a href=„<https://npo-echelon.ru/en/solutions/appchecker.php>“ rel=„nofollow“>AppChecker ©️ - Static analysis for C/C++/C#, PHP and Java <a href=„<https://www.ptsecurity.com/ww-en/products/ai/>“ rel=„nofollow“>Application Inspector ©️ - Combined SAST, DAST, IAST security scanner for C#, PHP, Java, SQL languages AppScan ©️ - Commercial Static Code Analysis. Supports: Microsoft .NET Framework (C#, ASP.NET, VB.NET), ASP (JavaScript/VBScript), C/C++, COBOL, ColdFusion, JavaScript, JavaServer Pages (JSP), Java™ (including support for Android APIs), Perl, PHP, PL/SQL, T-SQL, Visual Basic 6 <a href=„<https://appscreener.us>“ rel=„nofollow“>APPscreener ©️ - Static code analysis for binary and source code - Java/Scala, PHP, Javascript, C#, PL/SQL, Python, T-SQL, C/C++, ObjectiveC/Swift, Visual Basic 6.0, Ruby, Delphi, ABAP, HTML5 and Solidity Axivion Bauhaus Suite ©️ - Tracks down error-prone code locations, style violations, cloned or dead code, cyclic dependencies and more for C/C++, C#/.NET, Java and Ada 83/Ada 95 <a href=„<https://www.checkmarx.com/technology/supported-coding-languages/>“ rel=„nofollow“>Checkmarx ©️ - Commercial Static Code Analysis which doesn't require pre-compilation. Supports: Android (Java), Apex and VisualForce, ASP, C#, C/C++, Go, Groovy, HTML5, Java, JavaScript, Node.js, Objective C, Perl, PhoneGap, PHP, Python, Ruby, Scala, Swift, VB.NET, VB6, VBScript <a href=„<https://coala.io/>“ rel=„nofollow“>coala - Language independent framework for creating code analysis - supports <a href=„<https://coala.io/languages>“ rel=„nofollow“>over 60 languages by default <a href=„<http://spinroot.com/cobra/>“ rel=„nofollow“>Cobra ©️ - Structural source code analyzer by NASA's Jet Propulsion Laboratory. Supports C, C++, Ada, and Python. <a href=„<https://github.com/groupon/codeburner>“>codeburner - Provides a unified interface to sort and act on the issues it finds <a href=„<https://codefactor.io>“ rel=„nofollow“>CodeFactor ©️ - Static Code Analysis for C#, C, C++, CoffeeScript, CSS, Groovy, GO, JAVA, JavaScript, Less, Python, Ruby, Scala, SCSS, TypeScript. <a href=„<https://submain.com/products/codeit.right.aspx>“ rel=„nofollow“>CodeIt.Right ©️ - CodeIt.Right™ provides a fast, automated way to ensure that your source code adheres to (your) predefined design and style guidelines as well as best coding practices. Supported languages: C#, VB.NET. <a href=„<https://github.com/xcatliu/cqc>“>cqc - Check your code quality for js, jsx, vue, css, less, scss, sass and styl files. <a href=„<https://github.com/microsoft/devskim>“>DevSkim - Regex-based static analysis tool for Visual Studio, VS Code, and Sublime Text - C/C++, C#, PHP, ASP, Python, Ruby, Java, and others. <a href=„<https://software.microfocus.com/en-us/products/static-code-analysis-sast/overview>“ rel=„nofollow“>Fortify ©️ A commercial static analysis platform that supports the scanning of C/C++, C#, VB.NET, VB6, ABAP/BSP, ActionScript, Apex, ASP.NET, Classic ASP, VB Script, Cobol, ColdFusion, HTML, Java, JS, JSP, MXML/Flex, Objective-C, PHP, PL/SQL, T-SQL, Python (2.6, 2.7), Ruby (1.9.3), Swift, Scala, VB, and XML. <a href=„<https://github.com/sideci/goodcheck>“>Goodcheck - Regexp based customizable

linter

- <https://github.com/wireghoul/graudit> - Grep rough audit - source code auditing tool - C/C++, PHP, ASP, C#, Java, Perl, Python, Ruby
- <https://houndci.com/> - Comments on style violations in GitHub pull requests. Supports Coffeescript, Go, HAML, JavaScript, Ruby, SCSS and Swift.
- <https://github.com/justinabrahms/imhotep> - Comment on commits coming into your repository and check for syntactic errors and general lint warnings.
- <https://github.com/facebook/infer> - A static analyzer for Java, C and Objective-C
- <http://www.klocwork.com/products-services/klocwork> - Quality and Security Static analysis for C/C++, Java and C#
- <https://www.kiuwan.com/code-security-sast/> - Identify and remediate cyber threats in a blazingly fast, collaborative environment, with seamlessly integration in your SDLC. Python, C/C++, Java, C#, PHP and more
- <https://github.com/oclint/oclint> - A static source code analysis tool to improve quality and reduce defects for C, C++ and Objective-C
- <https://github.com/facebook/pfff> - Facebook's tools for code analysis, visualizations, or style-preserving source transformation for many languages
- <https://pmd.github.io/> - PMD - A source code analyzer for Java, Javascript, PLSQL, XML, XSL and others
- <https://github.com/prontolabs/pronto> - Pronto - Quick automated code review of your changes. Supports more than 40 runners for various languages, including Clang, Elixir, JavaScript, PHP, Ruby and more
- <https://github.com/pre-commit/pre-commit> - pre-commit - A framework for managing and maintaining multi-language pre-commit hooks.
- <https://github.com/PositiveTechnologies/PT.PM> - PT.PM - An engine for searching patterns in the source code, based on Unified AST or UST. At present time C#, Java, PHP, PL/SQL, T-SQL, and JavaScript are supported. Patterns can be described within the code or using a DSL.
- <https://www.viva64.com/en/pvs-studio/> - PVS-Studio - a (<https://www.viva64.com/en/b/0457/>) conditionally free for FOSS) static analysis of C/C++ and C# code. For advertising purposes <https://github.com/viva64/pvs-studio-check-list> you can propose a large FOSS project for analysis by PVS employees.
- <https://security-code-scan.github.io/> - Security Code Scan - Security code analyzer for C# and VB.NET. Detects various security vulnerability patterns: SQLi, XSS, CSRF, XXE, Open Redirect, etc.
- <https://semml.com/> - Semml QL and LGTM - Find security vulnerabilities, variants, and critical code quality issues using queries over source code. Automatic PR code review; free for public GitHub/Bitbucket repo: <https://LGTM.com>
- <https://github.com/google/shipshape> - shipshape - Static program analysis platform that allows custom analyzers to plug in through a common interface
- <http://www.sonarqube.org/> - SonarQube - SonarQube is an open platform to manage code quality.
- <https://github.com/StanfordPL/stoke> - STOKe - a programming-language agnostic stochastic optimizer for the x86_64 instruction set. It uses random search to explore the extremely high-dimensional space of all possible program transformations
- <https://www.synopsys.com/software-integrity/security-testing/static-analysis-sast.html> - Synopsys - A commercial static analysis platform that allows for scanning of multiple languages (C/C++, Android, C#, Java, JS, PHP, Python, Node.js, Ruby, Fortran, and Swift)
- <https://github.com/Tencent/TscanCode> - TscanCode - A fast and accurate static analysis solution for C/C++, C#, Lua codes provided by Tencent. Using GPLv3 license.
- <https://github.com/Yelp/undebt> - Undebt - Language-independent tool for massive, automatic, programmable refactoring based on simple pattern definitions
- <http://www.veracode.com/products/static-analysis-sast/static-code-analysis>

rel=„nofollow“>Veracode ©️ - Find flaws in binaries and bytecode without requiring source. Support all major programming languages: Java, .NET, JavaScript, Swift, Objective-C, C, C++ and more. WALA - static analysis capabilities for Java bytecode and related languages and for JavaScript Wotan - Pluggable TypeScript and JavaScript linter XCode ©️ - XCode provides a pretty decent UI for Clang's static code analyzer (C/C++, Obj-C) <h2><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Build tools</h2> checkmake - Linter / Analyzer for Makefiles codechecker - a defect database and viewer extension for the Clang Static Analyzer <h2><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Binaries</h2> BinSkim - A binary static analysis tool that provides security and correctness results for Windows portable executables. Jakstab - Jakstab is an Abstract Interpretation-based, integrated disassembly and static analysis framework for designing analyses on executables and recovering reliable control flow graphs. Manalyze - A static analyzer, which checks portable executables for malicious content. Twiggy - Analyzes a binary's call graph to profile code size. The goal is to slim down binaries. <h2><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Containers</h2> anchore - Discover, analyze, and certify container images clair - Vulnerability Static Analysis for Containers collector - Run arbitrary scripts inside containers, and gather useful information dagda - Perform static analysis of known vulnerabilities in docker images/containers. Docker Label Inspector - Lint and validate Dockerfile labels Haskell Dockerfile Linter - A smarter Dockerfile linter that helps you build best practice Docker images <h2><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Config Files</h2> gixy - a tool to analyze Nginx configuration. The main goal is to prevent misconfiguration and automate flaw detection. <h2><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Configuration Management</h2> ansible-lint - Checks playbooks for practices and behaviour that could potentially be improved <a

https://github.com/stelligent/cfn_nag - A linter for AWS CloudFormation templates.

<http://www.foodcritic.io/> - A lint tool that checks Chef cookbooks for common problems.

<https://github.com/rodjek/puppet-lint> - Puppet Lint - Check that your Puppet manifests conform to the style guide.

<https://github.com/mre/awesome-static-analysis#css> - CSS Stats - Potentially interesting stats on stylesheets

<https://github.com/csscomb/csscomb.js> - CSScomb - a coding style formatter for CSS. Supports own configurations to make style sheets beautiful and consistent

<https://github.com/CSSLint/csslint> - CSSLint - Does basic syntax checking and finds problematic patterns or signs of inefficiency

<https://github.com/katiefenn/parker> - Parker - Stylesheet analysis tool

<https://github.com/sasstools/sass-lint> - sass-lint - A Node-only Sass linter for both sass and scss syntax.

<https://github.com/brigade/scss-lint> - scsslint - Linter for SCSS files

<https://github.com/pocketjoso/specificity-graph> - CSS Specificity Graph Generator

<http://stylelint.io/> - Stylelint - Linter for SCSS/CSS files

<https://github.com/mre/awesome-static-analysis#gherkin> - Gherkin - A linter for the Gherkin-Syntax written in Javascript.

<https://github.com/mre/awesome-static-analysis#html> - HTML Inspector - HTML Inspector is a code quality tool to help you and your team write better markup.

<http://www.html-tidy.org/> - HTML Tidy - Corrects and cleans up HTML and XML documents by fixing markup errors and upgrading legacy code to modern standards.

<https://github.com/yaniswang/HTMLHint> - HTMLHint - A Static Code Analysis Tool for HTML

<https://github.com/Polymer/polymer-analyzer> - Polymer-analyzer - A static analysis framework for Web Components.

<https://github.com/mre/awesome-static-analysis#ide-plugins> - IDE Plugins

<https://github.com/w0rp/ale> - ale - Asynchronous Lint Engine for Vim and NeoVim with support for many languages

<https://www.attackflow.com/Extension> - Attackflow Extension - Attackflow plugin for Visual Studio, which enables developers to find critical security bugs at real time in the source code without any prior knowledge.

<https://github.com/Microsoft/DevSkim> - DevSkim - Inline, realtime security analysis. Works with multiple programming languages and IDEs (VS, VS Code, Sublime Text, ...).

<https://github.com/pumasecurity/puma-scan> - Puma Scan - Puma Scan provides real time secure code analysis for common vulnerabilities (XSS, SQLi, CSRF, LDAPi, crypto, deserialization, etc.) as development teams write code in Visual Studio.

<https://security-code-scan.github.io/> - Security Code Scan - Security code analyzer for C# and VB.NET that integrates into Visual Studio 2015 and newer. Detects various security vulnerability patterns: SQLi, XSS, CSRF, XXE, Open Redirect, etc.

<https://github.com/Kuniwak/vint> - vint - Fast and Highly Extensible Vim script Language

Lint implemented by Python.

- <https://github.com/mre/awesome-static-analysis#latex>

<https://github.com/mre/awesome-static-analysis#latex>
- <http://www.nongnu.org/chktex/>

<http://www.nongnu.org/chktex/>
- <http://www.ctan.org/pkg/lacheck>

<http://www.ctan.org/pkg/lacheck>

lacheck - A tool for finding common mistakes in LaTeX documents.

- <https://github.com/mre/awesome-static-analysis#makefiles>

<https://github.com/mre/awesome-static-analysis#makefiles>
- <https://www.freebsd.org/cgi/man.cgi?query=portlint&sektion=1&manpath=FreeBSD+8.1-RELEASE+and+Ports>

<https://www.freebsd.org/cgi/man.cgi?query=portlint&sektion=1&manpath=FreeBSD+8.1-RELEASE+and+Ports>

portlint - A verifier for FreeBSD and DragonFlyBSD port directories

- <https://github.com/mre/awesome-static-analysis#markdown>

<https://github.com/mre/awesome-static-analysis#markdown>
- <https://github.com/mivok/markdownlint>

<https://github.com/mivok/markdownlint>

mdlint - A tool to check markdown files and flag style issues.

- <https://github.com/mre/awesome-static-analysis#mobile>

<https://github.com/mre/awesome-static-analysis#mobile>
- <https://github.com/passy/android-lint-summary>

<https://github.com/passy/android-lint-summary>
- <https://github.com/secure-software-engineering/soot-infoflow-android>

<https://github.com/secure-software-engineering/soot-infoflow-android>
- <https://github.com/GeoffreyHecht/paprika>

<https://github.com/GeoffreyHecht/paprika>
- <https://github.com/linkedin/qark>

<https://github.com/linkedin/qark>

qark - Tool to look for several security related Android application vulnerabilities

- <https://github.com/mre/awesome-static-analysis#packages>

<https://github.com/mre/awesome-static-analysis#packages>
- <https://github.com/Debian/lintian>

<https://github.com/Debian/lintian>
- <https://github.com/rpm-software-management/rpmlint>

<https://github.com/rpm-software-management/rpmlint>

rpmlint - Tool for checking common errors in rpm packages

- <https://github.com/mre/awesome-static-analysis#supporting-tools>

<https://github.com/mre/awesome-static-analysis#supporting-tools>
- <https://github.com/uni-bremen-agst/libvcs4j>

<https://github.com/uni-bremen-agst/libvcs4j>

LibVCS4j - A Java library that allows existing tools to analyse the evolution of software systems by providing a common API for different version control systems and issue trackers.

- <https://github.com/mre/awesome-static-analysis#template-languages>

<https://github.com/mre/awesome-static-analysis#template-languages>
- <https://github.com/rwjblue/ember-template-lint>

<https://github.com/rwjblue/ember-template-lint>
- <https://github.com/brigade/haml-lint>

<https://github.com/brigade/haml-lint>
- <https://github.com/sds/slim-lint>

<https://github.com/sds/slim-lint>
- <https://github.com/adrienverge/yamllint>

<https://github.com/adrienverge/yamllint>

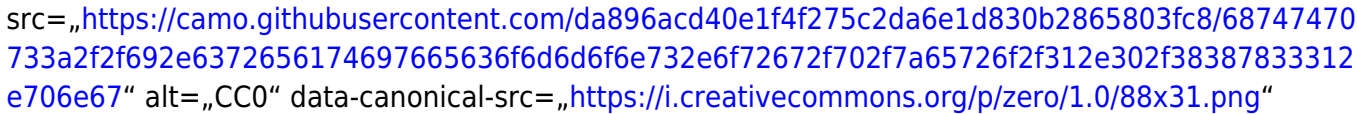
yamllint - Checks YAML files for syntax validity, key repetition and cosmetic problems such as lines length, trailing spaces, and indentation.

<h2><a id=„user-content-translation“ class=„anchor“ aria-hidden=„true“ href=„<https://github.com/mre/awesome-static-analysis#translation>“><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Translation</h2> <a href=„<https://github.com/willkg/dennis/>“>dennis - A set of utilities for working with PO files to ease development and improve quality. <h2><a id=„user-content-writing“ class=„anchor“ aria-hidden=„true“ href=„<https://github.com/mre/awesome-static-analysis#writing>“><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Writing</h2> <a href=„<https://github.com/vlajos/misspell-fixer>“>misspell-fixer - Quick tool for fixing common misspellings, typos in source code <a href=„<https://github.com/amperser/proselint/>“>proselint - a linter for English prose with a focus on writing style instead of grammar. <a href=„<https://github.com/ValeLint/vale>“>vale - A customizable, syntax-aware linter for prose. <a href=„<https://github.com/btford/write-good>“>write-good - A linter with a focus on eliminating „weasel words“. <h2><a id=„user-content-web-services“ class=„anchor“ aria-hidden=„true“ href=„<https://github.com/mre/awesome-static-analysis#web-services>“><svg class=„octicon octicon-link“ viewBox=„0 0 16 16“ version=„1.1“ width=„16“ height=„16“ aria-hidden=„true“/>Web services</h2> <a href=„<https://www.codacy.com/>“ rel=„nofollow“>Codacy ©️ - Code Analysis to ship Better Code, Faster. <a href=„<https://codeclimate.com/>“ rel=„nofollow“>Code Climate ©️ - The open and extensible static analysis platform, for everyone. <a href=„<https://codefactor.io>“ rel=„nofollow“>CodeFactor ©️ - Automated Code Analysis for repos on GitHub or BitBucket. <a href=„<https://mygamma.io>“ rel=„nofollow“>Gamma ©️ - An intelligent software analytics platform that identifies issues from multiple lenses: Design issues, code issues, duplication and metrics. Available for Java, C, C++ and C#. <a href=„<https://www.kiuwan.com/>“ rel=„nofollow“>kiuwan ©️ - Software Analytics in the Cloud supporting more than 22 programming languages. <a href=„<https://landscape.io/>“ rel=„nofollow“>Landscape ©️ - Static code analysis for Python <a href=„<https://layeredinsight.com/>“ rel=„nofollow“>Layered Insight ©️ - Container native application protection to provide visibility and control of containerized applications. <a href=„<https://lgtm.com/>“ rel=„nofollow“>LGTM.com ©️ - Deep code analysis for GitHub and Bitbucket to find security vulnerabilities and critical code quality issues (using Semmle QL). Automatic code review for pull requests; free for public repositories. <a href=„<https://nitpick-ci.com>“ rel=„nofollow“>Nitpick CI ©️ - Automated PHP code review <a href=„<https://www.quantifiedcode.com/>“ rel=„nofollow“>QuantifiedCode - Automated code review & repair <a href=„<https://scrutinizer-ci.com/>“ rel=„nofollow“>Scrutinizer ©️ - A proprietary code quality checker that can be integrated with GitHub <a href=„<https://insight.sensiolabs.com/>“ rel=„nofollow“>SensioLabs Insight ©️ - Detect security risks, find bugs and provide actionable metrics for PHP projects <a href=„<https://sider.review>“ rel=„nofollow“>Sider ©️ - An automated code reviewing tool. Improving developers' productivity. <a href=„<https:// snyk.io/>“ rel=„nofollow“>Snyk ©️ - Vulnerability scanner for dependencies of node.js apps (free for Open Source Projects) <a href=„<http://www.teamscale.com/>“ rel=„nofollow“>Teamscale ©️ - Static and dynamic analysis tool supporting more than 25 languages and direct IDE integration. Free hosting for Open Source projects available on request. Free academic licenses available. <a href=„<https://www.jetbrains.com/upsources/>“ rel=„nofollow“>Upsource ©️ - Code review tool with static code analysis and code-aware navigation for Java, PHP, JavaScript and

Kotlin.

- <https://github.com/dominikh/go-tools> - A collection of tools and libraries for working with Go code, including linters and static analysis
- <https://github.com/mcandre/linters> - An introduction to static code analysis
- <https://github.com/exakat/php-static-analysis-tools> - A reviewed list of useful PHP static analysis tools
- https://www.peerlyst.com/posts/a-list-of-static-analysis-tools-for-c-c-peerlyst?utm_source=twitter&utm_medium=social&utm_content=peerlyst_post&utm_campaign=peerlyst_resources - Tools for C/C++
- http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis - Wikipedia - A list of tools for static code analysis.

[License](https://github.com/mre/awesome-static-analysis#license)

<https://creativecommons.org/publicdomain/zero/1.0/> 

To the extent possible under law, <http://matthias-endler.de> Matthias Endler has waived all copyright and related or neighboring rights to this work. Title image <http://www.freepik.com> Designed by Freepik.

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

https://schnipsl.qgelm.de/doku.php?id=wallabag:mre_awesome-static-analysis

Last update: **2021/12/06 15:24**

