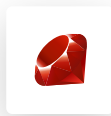


- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby**
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



## Ruby static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your RUBY code

All rules 42

Bug 7

Security Hotspot 2

Code Smell 33

Tags

Search by name...



All code should be reachable



Variables should not be self-assigned



Useless "if true ..." and "if false ..." blocks should be removed



Methods should not have identical implementations



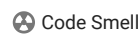
Two branches in a conditional structure should not have exactly the same implementation



"case" statements should not have too many "when" clauses



Unused function parameters should be removed



Track uses of "FIXME" tags



Redundant pairs of parentheses should be removed



Nested blocks of code should not be left empty



Functions should not have too many parameters



Collapsible "if" statements should be merged

"=+" should not be used instead of "+="

Analyze your code



The use of operators pairs ( +=, -= or != ) where the reversed, single operator was meant ( +=, -= or != ) will compile and run, but not produce the expected results.

This rule raises an issue when +=, -=, or != is used without any spacing between the two operators and when there is at least one whitespace character after.

Noncompliant Code Example

```
target = -5
num = 3

target -= num # Noncompliant; target = -3. Is that real
target += num # Noncompliant; target = 3
```

Compliant Solution


```
target = -5
num = 3

target = -num # Compliant; intent to assign inverse val
target += num
```


Available In:

sonarlint | sonarcloud | sonarqube


© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. [Privacy Policy](#)

 Code Smell


Using hardcoded IP addresses is security-sensitive

 Security Hotspot


Multi-line comments should not be empty

 Code Smell

Boolean checks should not be inverted

 Code Smell

Unused local variables should be removed

 Code Smell