sonar RULES

Products

- 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...

Methods should not have identical implementations

Code Smell

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

Code Smell

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

Code Smell

Unused function parameters should be removed

Code Smell

Track uses of "FIXME" tags

Code Smell

Redundant pairs of parentheses should be removed

Code Smell

Nested blocks of code should not be left empty

Code Smell

Functions should not have too many parameters

Code Smell

Collapsible "if" statements should be merged

Code Smell

Using hardcoded IP addresses is security-sensitive

Security Hotspot

Multi-line comments should not be empty

Code Smell

Related "if/elsif" statements and "when" in a "case" should not have the same condition

Analyze your code

Bug | Major | unused pitfall

A `case` and a chain of `if`/`elsif` statements is evaluated from top to bottom. At most, only one branch will be executed: the first one with a condition that evaluates to `true`.

Therefore, duplicating a condition automatically leads to dead code. Usually, this is due to a copy/paste error. At best, it's simply dead code and at worst, it's a bug that is likely to induce further bugs as the code is maintained, and obviously it could lead to unexpected behavior.

For a `case`, the second `when` will never be executed, rendering it dead code. Worse there is the risk in this situation that future maintenance will be done on the dead case, rather than on the one that's actually used.

Noncompliant Code Example

```
if param == 1
  openWindow()
elsif param == 2
  closeWindow()
elsif param == 1  # Noncompliant
  moveWindowToTheBackground()
end

case i
  when 1
    # ...
  when 3
    # ...
  when 1  # Noncompliant
    # ...
  else
    # ...
end
```

Compliant Solution

```
if param == 1
  openWindow()
elsif param == 2
  closeWindow()
elsif param == 3
  moveWindowToTheBackground()
end

case i
  when 1
    # ...
```

Boolean checks should not be inverted

Code Smell

Unused local variables should be removed

Code Smell

function and block parameter names should comply with a naming convention

Code Smell

Class names should comply with a naming convention

Code Smell

```
  when 3
    # ...
  else
    # ...
end
```

Available In:

sonarlint | sonarcloud | sonarqube