- 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...

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

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

Method names should comply with a naming convention

Code Smell

Track uses of "TODO" tags

Code Smell

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

Analyze your code

Bug | Major | cwe

`if` statements with conditions that are always false have the effect of making blocks of code non-functional. `if` statements with conditions that are always true are completely redundant, and make the code less readable.

There are three possible causes for the presence of such code:

- An if statement was changed during debugging and that debug code has been committed.
- Some value was left unset.
- Some logic is not doing what the programmer thought it did.

In any of these cases, unconditional `if` statements should be removed.

Noncompliant Code Example

```
if true
  doSomething()
end
...
if false
  doSomethingElse()
end
```

Compliant Solution

```
doSomething()
...
```

See

- MITRE, CWE-489 - Active Debug Code
- MITRE, CWE-570 - Expression is Always False
- MITRE, CWE-571 - Expression is Always True

Available In:

sonarlint | sonarcloud | sonarqube

Track lack of copyright and license headers

Code Smell

Octal values should not be used

Code Smell

"case" statements should not be nested

Code Smell

Control flow statements "if", "for", "while", "until", "case" and "begin...rescue" should not be nested too deeply

Code Smell