Ruby static code analysis

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

1. Hard-coded credentials are security-sensitive
   Security Hotspot
2. Cognitive Complexity of functions should not be too high
   Code Smell
3. "case" statements should have "else" clauses
   Code Smell
4. String literals should not be duplicated
   Code Smell
5. Methods should not be empty
   Code Smell
6. All branches in a conditional structure should not have exactly the same implementation
   Bug
7. "=+" should not be used instead of "+="
   Bug
8. Related "if/elsif" statements and "when" in a "case" should not have the same condition
   Bug
9. Identical expressions should not be used on both sides of a binary operator
   Bug
10. All code should be reachable
    Bug
11. Variables should not be self-assigned
    Bug
12. Useless "if true ..." and "if false ..." blocks should be removed
    Bug
13. Methods should not have identical implementations
    Code Smell
14. Two branches in a conditional structure should not have exactly the same implementation
    Code Smell
15. "case" statements should not have too many "when" clauses
    Code Smell

16.

Unused function parameters should be removed
Code Smell

17.

Track uses of "FIXME" tags
Code Smell

18.

Redundant pairs of parentheses should be removed
Code Smell

19.

Nested blocks of code should not be left empty
Code Smell

20.

Functions should not have too many parameters
Code Smell

21.

Collapsible "if" statements should be merged
Code Smell

22.

Using hardcoded IP addresses is security-sensitive
Security Hotspot

23.

Multi-line comments should not be empty
Code Smell

24.

Boolean checks should not be inverted
Code Smell

25.

Unused local variables should be removed
Code Smell

26.

function and block parameter names should comply with a naming convention
Code Smell

27.

Class names should comply with a naming convention
Code Smell

28.

Method names should comply with a naming convention
Code Smell

29.

Track uses of "TODO" tags
Code Smell

30.

Track lack of copyright and license headers
Code Smell

31.

Octal values should not be used
Code Smell

32.

"case" statements should not be nested
Code Smell

33.

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

34.

"if ... else if" constructs should end with "else" clauses
Code Smell

35.

Expressions should not be too complex
Code Smell

36.

Ruby parser failure
Code Smell

37.

Functions should not have too many lines of code
Code Smell

38.

Statements should be on separate lines
Code Smell

39.

"case when" clauses should not have too many lines of code
Code Smell

40.

Files should not have too many lines of code
Code Smell

41.

Lines should not be too long
Code Smell

42.

Tabulation characters should not be used
Code Smell