**C static code analysis**

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

### ["memset" should not be used to delete sensitive data](https://rules.sonarsource.com/c/RSPEC-5798)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5798)

### [POSIX functions should not be called with arguments that trigger buffer overflows](https://rules.sonarsource.com/c/RSPEC-5782)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5782)

### [XML parsers should not be vulnerable to XXE attacks](https://rules.sonarsource.com/c/RSPEC-2755)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-2755)

### [Function-like macros should not be invoked without all of their arguments](https://rules.sonarsource.com/c/RSPEC-961)

[Bug](https://rules.sonarsource.com/c/RSPEC-961)

### [The address of an automatic object should not be assigned to another object that may persist after the first object has ceased to exist](https://rules.sonarsource.com/c/RSPEC-946)

[Bug](https://rules.sonarsource.com/c/RSPEC-946)

### ["pthread\_mutex\_t" should be unlocked in the reverse order they were locked](https://rules.sonarsource.com/c/RSPEC-5489)

[Bug](https://rules.sonarsource.com/c/RSPEC-5489)

### ["pthread\_mutex\_t" should be properly initialized and destroyed](https://rules.sonarsource.com/c/RSPEC-5487)

[Bug](https://rules.sonarsource.com/c/RSPEC-5487)

### ["pthread\_mutex\_t" should not be consecutively locked or unlocked twice](https://rules.sonarsource.com/c/RSPEC-5486)

[Bug](https://rules.sonarsource.com/c/RSPEC-5486)

### [Functions with "noreturn" attribute should not return](https://rules.sonarsource.com/c/RSPEC-5267)

[Bug](https://rules.sonarsource.com/c/RSPEC-5267)

### ["memcmp" should only be called with pointers to trivially copyable types with no padding](https://rules.sonarsource.com/c/RSPEC-5000)

[Bug](https://rules.sonarsource.com/c/RSPEC-5000)

### [Stack allocated memory and non-owned memory should not be freed](https://rules.sonarsource.com/c/RSPEC-3590)

[Bug](https://rules.sonarsource.com/c/RSPEC-3590)

### [Closed resources should not be accessed](https://rules.sonarsource.com/c/RSPEC-3588)

[Bug](https://rules.sonarsource.com/c/RSPEC-3588)

### [Dynamically allocated memory should be released](https://rules.sonarsource.com/c/RSPEC-3584)

[Bug](https://rules.sonarsource.com/c/RSPEC-3584)

### [Freed memory should not be used](https://rules.sonarsource.com/c/RSPEC-3529)

[Bug](https://rules.sonarsource.com/c/RSPEC-3529)

### [Memory locations should not be released more than once](https://rules.sonarsource.com/c/RSPEC-3520)

[Bug](https://rules.sonarsource.com/c/RSPEC-3520)

### [Memory access should be explicitly bounded to prevent buffer overflows](https://rules.sonarsource.com/c/RSPEC-3519)

[Bug](https://rules.sonarsource.com/c/RSPEC-3519)

### [Printf-style format strings should not lead to unexpected behavior at runtime](https://rules.sonarsource.com/c/RSPEC-2275)

[Bug](https://rules.sonarsource.com/c/RSPEC-2275)

### [Recursion should not be infinite](https://rules.sonarsource.com/c/RSPEC-2190)

[Bug](https://rules.sonarsource.com/c/RSPEC-2190)

### [Resources should be closed](https://rules.sonarsource.com/c/RSPEC-2095)

[Bug](https://rules.sonarsource.com/c/RSPEC-2095)

### [Hard-coded credentials are security-sensitive](https://rules.sonarsource.com/c/RSPEC-2068)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-2068)

### ["goto" should jump to labels declared later in the same function](https://rules.sonarsource.com/c/RSPEC-999)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-999)

### [Only standard forms of the "defined" directive should be used](https://rules.sonarsource.com/c/RSPEC-969)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-969)

### [Switch labels should not be nested inside non-switch blocks](https://rules.sonarsource.com/c/RSPEC-916)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-916)

### [The right-hand operands of && and || should not contain side effects](https://rules.sonarsource.com/c/RSPEC-912)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-912)

### [Digraphs should not be used](https://rules.sonarsource.com/c/RSPEC-798)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-798)

### [Trigraphs should not be used](https://rules.sonarsource.com/c/RSPEC-797)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-797)

### ["case" ranges should cover multiple values](https://rules.sonarsource.com/c/RSPEC-3936)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3936)

### [Array indices should be placed between brackets](https://rules.sonarsource.com/c/RSPEC-3729)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3729)

### [Redundant pointer operator sequences should be removed](https://rules.sonarsource.com/c/RSPEC-3491)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3491)

### [Non-reentrant POSIX functions should be replaced with their reentrant versions](https://rules.sonarsource.com/c/RSPEC-1912)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1912)

### ["goto" statements should not be used to jump into blocks](https://rules.sonarsource.com/c/RSPEC-1909)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1909)

### [Keywords introduced in later specifications should not be used as identifiers](https://rules.sonarsource.com/c/RSPEC-1760)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1760)

### [Switch cases should end with an unconditional "break" statement](https://rules.sonarsource.com/c/RSPEC-128)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-128)

### ["switch" statements should not contain non-case labels](https://rules.sonarsource.com/c/RSPEC-1219)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1219)

### [Control should not be transferred into a complex logic block using a "goto" or a "switch" statement](https://rules.sonarsource.com/c/RSPEC-1036)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1036)

### [Accessing files should not introduce TOCTOU vulnerabilities](https://rules.sonarsource.com/c/RSPEC-5847)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5847)

### [Cipher algorithms should be robust](https://rules.sonarsource.com/c/RSPEC-5547)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5547)

### [Encryption algorithms should be used with secure mode and padding scheme](https://rules.sonarsource.com/c/RSPEC-5542)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5542)

### [Server hostnames should be verified during SSL/TLS connections](https://rules.sonarsource.com/c/RSPEC-5527)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5527)

### [Server certificates should be verified during SSL/TLS connections](https://rules.sonarsource.com/c/RSPEC-4830)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-4830)

### [Cryptographic keys should be robust](https://rules.sonarsource.com/c/RSPEC-4426)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-4426)

### [Weak SSL/TLS protocols should not be used](https://rules.sonarsource.com/c/RSPEC-4423)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-4423)

### [Insecure functions should not be used](https://rules.sonarsource.com/c/RSPEC-1081)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-1081)

### ["scanf()" and "fscanf()" format strings should specify a field width for the "%s" string placeholder](https://rules.sonarsource.com/c/RSPEC-1079)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-1079)

### [Function exit paths should have appropriate return values](https://rules.sonarsource.com/c/RSPEC-935)

[Bug](https://rules.sonarsource.com/c/RSPEC-935)

### ["volatile" should not be used to qualify objects for which the meaning is not defined](https://rules.sonarsource.com/c/RSPEC-6200)

[Bug](https://rules.sonarsource.com/c/RSPEC-6200)

### [Relational and subtraction operators should not be used with pointers to different arrays](https://rules.sonarsource.com/c/RSPEC-5658)

[Bug](https://rules.sonarsource.com/c/RSPEC-5658)

### [Arguments evaluation order should not be relied on](https://rules.sonarsource.com/c/RSPEC-5570)

[Bug](https://rules.sonarsource.com/c/RSPEC-5570)

### [Parameter values should be appropriate](https://rules.sonarsource.com/c/RSPEC-3807)

[Bug](https://rules.sonarsource.com/c/RSPEC-3807)

### [Zero should not be a possible denominator](https://rules.sonarsource.com/c/RSPEC-3518)

[Bug](https://rules.sonarsource.com/c/RSPEC-3518)

### [Line-splicing should not be used in "//" comments](https://rules.sonarsource.com/c/RSPEC-2323)

[Bug](https://rules.sonarsource.com/c/RSPEC-2323)

### [Pointers should not be cast to integral types](https://rules.sonarsource.com/c/RSPEC-1767)

[Bug](https://rules.sonarsource.com/c/RSPEC-1767)

### ["sprintf" should not be used](https://rules.sonarsource.com/c/RSPEC-6069)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-6069)

### [Changing working directories without verifying the success is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5982)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5982)

### [Using "tmpnam", "tmpnam\_s" or "tmpnam\_r" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5824)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5824)

### [Changing directories improperly when using "chroot" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5802)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5802)

### [Using publicly writable directories is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5443)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5443)

### [Using clear-text protocols is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5332)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5332)

### [Expanding archive files without controlling resource consumption is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5042)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5042)

### [Using weak hashing algorithms is security-sensitive](https://rules.sonarsource.com/c/RSPEC-4790)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-4790)

### [Using pseudorandom number generators (PRNGs) is security-sensitive](https://rules.sonarsource.com/c/RSPEC-2245)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-2245)

### ["#undef" should be used with caution](https://rules.sonarsource.com/c/RSPEC-959)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-959)

### [Function names should be used either as a call with a parameter list or with the "&" operator](https://rules.sonarsource.com/c/RSPEC-936)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-936)

### [Functions should not be defined with a variable number of arguments](https://rules.sonarsource.com/c/RSPEC-923)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-923)

### [A cast shall not remove any const or volatile qualification from the type of a pointer or reference](https://rules.sonarsource.com/c/RSPEC-859)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-859)

### [Object and function types should be explicitly stated in their declarations and definitions](https://rules.sonarsource.com/c/RSPEC-820)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-820)

### [Functions should be declared explicitly](https://rules.sonarsource.com/c/RSPEC-819)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-819)

### [Appropriate arguments should be passed to UNIX/POSIX functions](https://rules.sonarsource.com/c/RSPEC-5488)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5488)

### [Appropriate arguments should be passed to stream functions](https://rules.sonarsource.com/c/RSPEC-5485)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5485)

### [Blocking functions should not be called inside critical sections](https://rules.sonarsource.com/c/RSPEC-5314)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5314)

### [Return value of "setuid" family of functions should always be checked](https://rules.sonarsource.com/c/RSPEC-5308)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5308)

### [Size of variable length arrays should be positive](https://rules.sonarsource.com/c/RSPEC-5283)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5283)

### [Argument of "printf" should be a format string](https://rules.sonarsource.com/c/RSPEC-5281)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5281)

### ["mktemp" family of functions templates should have at least six trailing "X"s](https://rules.sonarsource.com/c/RSPEC-5280)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5280)

### [Logical operators should not be confused with bitwise operators](https://rules.sonarsource.com/c/RSPEC-5263)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5263)

### [Header guards should be followed by according "#define" macro](https://rules.sonarsource.com/c/RSPEC-5259)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5259)

### ["default" clauses should be first or last](https://rules.sonarsource.com/c/RSPEC-4524)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-4524)

### [A conditionally executed single line should be denoted by indentation](https://rules.sonarsource.com/c/RSPEC-3973)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3973)

### [Conditionals should start on new lines](https://rules.sonarsource.com/c/RSPEC-3972)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3972)

### [Cognitive Complexity of functions should not be too high](https://rules.sonarsource.com/c/RSPEC-3776)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3776)

### [Control characters should not be used in literals](https://rules.sonarsource.com/c/RSPEC-2479)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2479)

### ["restrict" should not be used](https://rules.sonarsource.com/c/RSPEC-1836)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1836)

### ["static" should not be used for the size of an array parameter](https://rules.sonarsource.com/c/RSPEC-1831)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1831)

### [The sign of an unsigned variable should not be tested](https://rules.sonarsource.com/c/RSPEC-1768)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1768)

### [Pre-defined macros should not be defined, redefined or undefined](https://rules.sonarsource.com/c/RSPEC-1761)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1761)

### [Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply](https://rules.sonarsource.com/c/RSPEC-134)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-134)

### [Methods should not be empty](https://rules.sonarsource.com/c/RSPEC-1186)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1186)

### [Account validity should be verified when authenticating users with PAM](https://rules.sonarsource.com/c/RSPEC-5832)

[Vulnerability](https://rules.sonarsource.com/c/RSPEC-5832)

### [Lines starting with "#" should contain valid preprocessing directives](https://rules.sonarsource.com/c/RSPEC-977)

[Bug](https://rules.sonarsource.com/c/RSPEC-977)

### ["#include" directives should be followed by either <filename> or "filename" sequences](https://rules.sonarsource.com/c/RSPEC-956)

[Bug](https://rules.sonarsource.com/c/RSPEC-956)

### [Non-standard characters should not occur in header file names in "#include" directives](https://rules.sonarsource.com/c/RSPEC-955)

[Bug](https://rules.sonarsource.com/c/RSPEC-955)

### [Non-empty statements should change control flow or have at least one side-effect](https://rules.sonarsource.com/c/RSPEC-905)

[Bug](https://rules.sonarsource.com/c/RSPEC-905)

### [Unary minus should not be applied to an unsigned expression](https://rules.sonarsource.com/c/RSPEC-876)

[Bug](https://rules.sonarsource.com/c/RSPEC-876)

### [Objects with integer type should not be converted to objects with pointer type](https://rules.sonarsource.com/c/RSPEC-860)

[Bug](https://rules.sonarsource.com/c/RSPEC-860)

### [Variables should be initialized before use](https://rules.sonarsource.com/c/RSPEC-836)

[Bug](https://rules.sonarsource.com/c/RSPEC-836)

### [String literals with different prefixes should not be concatenated](https://rules.sonarsource.com/c/RSPEC-817)

[Bug](https://rules.sonarsource.com/c/RSPEC-817)

### [Only escape sequences defined in the ISO C standard should be used](https://rules.sonarsource.com/c/RSPEC-796)

[Bug](https://rules.sonarsource.com/c/RSPEC-796)

### ["#pragma pack" should be used correctly](https://rules.sonarsource.com/c/RSPEC-5501)

[Bug](https://rules.sonarsource.com/c/RSPEC-5501)

### [Enums should be consistent with the bit fields they initialize](https://rules.sonarsource.com/c/RSPEC-5491)

[Bug](https://rules.sonarsource.com/c/RSPEC-5491)

### [Array values should not be replaced unconditionally](https://rules.sonarsource.com/c/RSPEC-4143)

[Bug](https://rules.sonarsource.com/c/RSPEC-4143)

### [Integral operations should not overflow](https://rules.sonarsource.com/c/RSPEC-3949)

[Bug](https://rules.sonarsource.com/c/RSPEC-3949)

### ["case" ranges should not be empty](https://rules.sonarsource.com/c/RSPEC-3935)

[Bug](https://rules.sonarsource.com/c/RSPEC-3935)

### [All branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/c/RSPEC-3923)

[Bug](https://rules.sonarsource.com/c/RSPEC-3923)

### [Declaration specifiers should not be redundant](https://rules.sonarsource.com/c/RSPEC-3689)

[Bug](https://rules.sonarsource.com/c/RSPEC-3689)

### ["sizeof" should not be called on pointers](https://rules.sonarsource.com/c/RSPEC-3135)

[Bug](https://rules.sonarsource.com/c/RSPEC-3135)

### [Unary prefix operators should not be repeated](https://rules.sonarsource.com/c/RSPEC-2761)

[Bug](https://rules.sonarsource.com/c/RSPEC-2761)

### ["=+" should not be used instead of "+="](https://rules.sonarsource.com/c/RSPEC-2757)

[Bug](https://rules.sonarsource.com/c/RSPEC-2757)

### [Values of different "enum" types should not be compared](https://rules.sonarsource.com/c/RSPEC-2753)

[Bug](https://rules.sonarsource.com/c/RSPEC-2753)

### [Conditionally executed code should be reachable](https://rules.sonarsource.com/c/RSPEC-2583)

[Bug](https://rules.sonarsource.com/c/RSPEC-2583)

### [Null pointers should not be dereferenced](https://rules.sonarsource.com/c/RSPEC-2259)

[Bug](https://rules.sonarsource.com/c/RSPEC-2259)

### [Single-bit named bit fields should not be of a signed type](https://rules.sonarsource.com/c/RSPEC-2216)

[Bug](https://rules.sonarsource.com/c/RSPEC-2216)

### [Values should not be uselessly incremented](https://rules.sonarsource.com/c/RSPEC-2123)

[Bug](https://rules.sonarsource.com/c/RSPEC-2123)

### ["sizeof(sizeof(...))" should not be used](https://rules.sonarsource.com/c/RSPEC-1913)

[Bug](https://rules.sonarsource.com/c/RSPEC-1913)

### [Related "if/else if" statements should not have the same condition](https://rules.sonarsource.com/c/RSPEC-1862)

[Bug](https://rules.sonarsource.com/c/RSPEC-1862)

### [Identical expressions should not be used on both sides of a binary operator](https://rules.sonarsource.com/c/RSPEC-1764)

[Bug](https://rules.sonarsource.com/c/RSPEC-1764)

### [All code should be reachable](https://rules.sonarsource.com/c/RSPEC-1763)

[Bug](https://rules.sonarsource.com/c/RSPEC-1763)

### [Loops with at most one iteration should be refactored](https://rules.sonarsource.com/c/RSPEC-1751)

[Bug](https://rules.sonarsource.com/c/RSPEC-1751)

### [Variables should not be self-assigned](https://rules.sonarsource.com/c/RSPEC-1656)

[Bug](https://rules.sonarsource.com/c/RSPEC-1656)

### [Setting capabilities is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5849)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5849)

### [Using "strncpy" or "wcsncpy" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5816)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5816)

### [Using "strncat" or "wcsncat" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5815)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5815)

### [Using "strcat" or "wcscat" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5814)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5814)

### [Using "strlen" or "wcslen" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5813)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5813)

### [Using "strcpy" or "wcscpy" is security-sensitive](https://rules.sonarsource.com/c/RSPEC-5801)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-5801)

### [Setting loose POSIX file permissions is security-sensitive](https://rules.sonarsource.com/c/RSPEC-2612)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-2612)

### [#include directives in a file should only be preceded by other preprocessor directives or comments](https://rules.sonarsource.com/c/RSPEC-954)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-954)

### [Loops should not have more than one "break" or "goto" statement](https://rules.sonarsource.com/c/RSPEC-924)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-924)

### [Unused type declarations should be removed](https://rules.sonarsource.com/c/RSPEC-897)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-897)

### [Comma operator should not be used](https://rules.sonarsource.com/c/RSPEC-878)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-878)

### ["bool" expressions should not be used as operands to built-in operators other than =, &&, ||, !, ==, !=, unary &, and the conditional operator](https://rules.sonarsource.com/c/RSPEC-872)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-872)

### ["enum" members other than the first one should not be explicitly initialized unless all members are explicitly initialized](https://rules.sonarsource.com/c/RSPEC-841)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-841)

### [If a function has internal linkage then all re-declarations shall include the static storage class specifer](https://rules.sonarsource.com/c/RSPEC-833)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-833)

### [Functions should not be declared at block scope](https://rules.sonarsource.com/c/RSPEC-824)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-824)

### [Bit fields should be declared with appropriate types](https://rules.sonarsource.com/c/RSPEC-814)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-814)

### [Size of bit fields should not exceed the size of their types](https://rules.sonarsource.com/c/RSPEC-5494)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5494)

### [GNU attributes should be used correctly](https://rules.sonarsource.com/c/RSPEC-5297)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5297)

### [Unevaluated operands should not have side effects](https://rules.sonarsource.com/c/RSPEC-5279)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5279)

### [Size argument of memory functions should be consistent](https://rules.sonarsource.com/c/RSPEC-5278)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5278)

### [Implicit casts should not lower precision](https://rules.sonarsource.com/c/RSPEC-5276)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5276)

### [Appropriate size arguments should be passed to "strncat" and "strlcpy"](https://rules.sonarsource.com/c/RSPEC-5273)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5273)

### [Keywords shall not be used as macros identifiers](https://rules.sonarsource.com/c/RSPEC-5266)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5266)

### [Dereferenced null pointers should not be bound to references](https://rules.sonarsource.com/c/RSPEC-5262)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5262)

### ["else" statements should be clearly matched with an "if"](https://rules.sonarsource.com/c/RSPEC-5261)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5261)

### [Include directives should not rely on non-portable search strategy](https://rules.sonarsource.com/c/RSPEC-4263)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-4263)

### ["#include" paths should be portable](https://rules.sonarsource.com/c/RSPEC-3806)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3806)

### ["#import" should not be used](https://rules.sonarsource.com/c/RSPEC-3805)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3805)

### [Atomic types should be used instead of "volatile" types](https://rules.sonarsource.com/c/RSPEC-3687)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3687)

### ["switch" statements should cover all cases](https://rules.sonarsource.com/c/RSPEC-3562)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3562)

### [Methods returns should not be invariant](https://rules.sonarsource.com/c/RSPEC-3516)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3516)

### [Printf-style format strings should be used correctly](https://rules.sonarsource.com/c/RSPEC-3457)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3457)

### [Conditional operators should not be nested](https://rules.sonarsource.com/c/RSPEC-3358)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3358)

### [Multiline blocks should be enclosed in curly braces](https://rules.sonarsource.com/c/RSPEC-2681)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2681)

### [Increment should not be used to set boolean variables to 'true'](https://rules.sonarsource.com/c/RSPEC-2668)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2668)

### [Boolean expressions should not be gratuitous](https://rules.sonarsource.com/c/RSPEC-2589)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2589)

### [Parameters should be passed in the correct order](https://rules.sonarsource.com/c/RSPEC-2234)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2234)

### [Obsolete POSIX functions should not be used](https://rules.sonarsource.com/c/RSPEC-1911)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1911)

### [Two branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/c/RSPEC-1871)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1871)

### [Unused assignments should be removed](https://rules.sonarsource.com/c/RSPEC-1854)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1854)

### [Structures should not have too many fields](https://rules.sonarsource.com/c/RSPEC-1820)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1820)

### ["switch" statements should not have too many "case" clauses](https://rules.sonarsource.com/c/RSPEC-1479)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1479)

### [Sections of code should not be commented out](https://rules.sonarsource.com/c/RSPEC-125)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-125)

### [Deprecated K&R syntax should not be used for function definition](https://rules.sonarsource.com/c/RSPEC-1198)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1198)

### [Unused function parameters should be removed](https://rules.sonarsource.com/c/RSPEC-1172)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1172)

### [Unused functions and methods should be removed](https://rules.sonarsource.com/c/RSPEC-1144)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1144)

### [Track uses of "FIXME" tags](https://rules.sonarsource.com/c/RSPEC-1134)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1134)

### [Deprecated attributes should include explanations](https://rules.sonarsource.com/c/RSPEC-1123)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1123)

### [Assignments should not be made from within sub-expressions](https://rules.sonarsource.com/c/RSPEC-1121)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1121)

### [Variables should not be shadowed](https://rules.sonarsource.com/c/RSPEC-1117)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1117)

### [Redundant pairs of parentheses should be removed](https://rules.sonarsource.com/c/RSPEC-1110)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1110)

### [Nested blocks of code should not be left empty](https://rules.sonarsource.com/c/RSPEC-108)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-108)

### [Functions should not have too many parameters](https://rules.sonarsource.com/c/RSPEC-107)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-107)

### [Collapsible "if" statements should be merged](https://rules.sonarsource.com/c/RSPEC-1066)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1066)

### [Unused labels should be removed](https://rules.sonarsource.com/c/RSPEC-1065)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1065)

### [The "sizeof" and "alignof" operator should not be used with operands of a "void" type](https://rules.sonarsource.com/c/RSPEC-2665)

[Bug](https://rules.sonarsource.com/c/RSPEC-2665)

### ["nonnull" pointers should not be set to null](https://rules.sonarsource.com/c/RSPEC-2637)

[Bug](https://rules.sonarsource.com/c/RSPEC-2637)

### ["for" loop counters should not have essentially floating type](https://rules.sonarsource.com/c/RSPEC-2193)

[Bug](https://rules.sonarsource.com/c/RSPEC-2193)

### [Line continuation characters '\' should not be followed by trailing whitespace](https://rules.sonarsource.com/c/RSPEC-1916)

[Bug](https://rules.sonarsource.com/c/RSPEC-1916)

### [Using hardcoded IP addresses is security-sensitive](https://rules.sonarsource.com/c/RSPEC-1313)

[Security Hotspot](https://rules.sonarsource.com/c/RSPEC-1313)

### [Pointer and reference parameters should be "const" if the corresponding object is not modified](https://rules.sonarsource.com/c/RSPEC-995)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-995)

### [The three expressions of a "for" statement should only be concerned with loop control](https://rules.sonarsource.com/c/RSPEC-886)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-886)

### [Literal suffix "L" for long integers shall be upper case](https://rules.sonarsource.com/c/RSPEC-818)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-818)

### [Multicharacter literals should not be used](https://rules.sonarsource.com/c/RSPEC-5978)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5978)

### [Loop variables should be declared in the minimal possible scope](https://rules.sonarsource.com/c/RSPEC-5955)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5955)

### [Macros should not be used as replacement to "typdef" and "using"](https://rules.sonarsource.com/c/RSPEC-5825)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5825)

### ["^" should not be confused with exponentiation](https://rules.sonarsource.com/c/RSPEC-5381)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5381)

### [Pointer and reference local variables should be "const" if the corresponding object is not modified](https://rules.sonarsource.com/c/RSPEC-5350)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5350)

### [Format strings should comply with ISO standards](https://rules.sonarsource.com/c/RSPEC-5293)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5293)

### [Functions which do not return should be declared as "noreturn"](https://rules.sonarsource.com/c/RSPEC-5271)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5271)

### [Macros should not be redefined](https://rules.sonarsource.com/c/RSPEC-3744)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3744)

### ["#include\_next" should not be used](https://rules.sonarsource.com/c/RSPEC-3730)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3730)

### [String literals should not be concatenated implicitly](https://rules.sonarsource.com/c/RSPEC-3728)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3728)

### [Types and variables should be declared in separate statements](https://rules.sonarsource.com/c/RSPEC-3646)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3646)

### [Jump statements should not be redundant](https://rules.sonarsource.com/c/RSPEC-3626)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3626)

### [Empty "case" clauses that fall through to the "default" should be omitted](https://rules.sonarsource.com/c/RSPEC-3458)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3458)

### [Forward declarations should not be redundant](https://rules.sonarsource.com/c/RSPEC-3231)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3231)

### [Declarations should not be empty](https://rules.sonarsource.com/c/RSPEC-2754)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2754)

### [Redundant casts should not be used](https://rules.sonarsource.com/c/RSPEC-1905)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1905)

### [Code annotated as deprecated should not be used](https://rules.sonarsource.com/c/RSPEC-1874)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1874)

### ["#pragma warning (default: ...)" should not be used](https://rules.sonarsource.com/c/RSPEC-1762)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1762)

### [Init-declarator-lists and member-declarator-lists should consist of single init-declarators and member-declarators respectively](https://rules.sonarsource.com/c/RSPEC-1659)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1659)

### [Unused local variables should be removed](https://rules.sonarsource.com/c/RSPEC-1481)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1481)

### ["switch" statements should have at least 3 "case" clauses](https://rules.sonarsource.com/c/RSPEC-1301)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1301)

### [A "while" loop should be used instead of a "for" loop](https://rules.sonarsource.com/c/RSPEC-1264)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1264)

### [Nested code blocks should not be used](https://rules.sonarsource.com/c/RSPEC-1199)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1199)

### [Empty statements should be removed](https://rules.sonarsource.com/c/RSPEC-1116)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1116)

### ["/\*" and "//" should not be used within comments](https://rules.sonarsource.com/c/RSPEC-1103)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1103)

### [Track uses of "TODO" tags](https://rules.sonarsource.com/c/RSPEC-1135)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1135)

### [Deprecated code should be removed](https://rules.sonarsource.com/c/RSPEC-1133)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1133)

### [Reserved identifiers and functions in the C standard library should not be defined or declared](https://rules.sonarsource.com/c/RSPEC-978)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-978)

### [Bit fields should not be used](https://rules.sonarsource.com/c/RSPEC-2806)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2806)

### [Track lack of copyright and license headers](https://rules.sonarsource.com/c/RSPEC-1451)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1451)

### [Octal values should not be used](https://rules.sonarsource.com/c/RSPEC-1314)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1314)

### ["abort", "exit", "getenv" and "system" from <stdlib.h> should not be used](https://rules.sonarsource.com/c/RSPEC-990)

[Bug](https://rules.sonarsource.com/c/RSPEC-990)

### ["atof", "atoi" and "atol" from <stdlib.h> should not be used](https://rules.sonarsource.com/c/RSPEC-989)

[Bug](https://rules.sonarsource.com/c/RSPEC-989)

### ["<signal.h>" should not be used](https://rules.sonarsource.com/c/RSPEC-987)

[Bug](https://rules.sonarsource.com/c/RSPEC-987)

### [Dynamic heap memory allocation should not be used](https://rules.sonarsource.com/c/RSPEC-984)

[Bug](https://rules.sonarsource.com/c/RSPEC-984)

### ["<time.h>" should not be used](https://rules.sonarsource.com/c/RSPEC-991)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-991)

### ["<stdio.h>" should not be used in production code](https://rules.sonarsource.com/c/RSPEC-988)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-988)

### ["offsetof" macro from <stddef.h> should not be used](https://rules.sonarsource.com/c/RSPEC-986)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-986)

### ["errno" should not be used](https://rules.sonarsource.com/c/RSPEC-985)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-985)

### ["setjmp" and "longjmp" should not be used](https://rules.sonarsource.com/c/RSPEC-982)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-982)

### [Function-like macros should not be used](https://rules.sonarsource.com/c/RSPEC-960)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-960)

### [Macros should not be #define'd or #undef'd within a block](https://rules.sonarsource.com/c/RSPEC-958)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-958)

### [Unions should not be used](https://rules.sonarsource.com/c/RSPEC-953)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-953)

### [Object declarations should contain no more than 2 levels of pointer indirection](https://rules.sonarsource.com/c/RSPEC-943)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-943)

### [Functions without parameters should be declared with parameter type "void"](https://rules.sonarsource.com/c/RSPEC-929)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-929)

### [Recursion should not be used](https://rules.sonarsource.com/c/RSPEC-925)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-925)

### [Constants of unsigned type should have a "U" suffix](https://rules.sonarsource.com/c/RSPEC-854)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-854)

### [Octal and hexadecimal escape sequences should be terminated](https://rules.sonarsource.com/c/RSPEC-2335)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2335)

### [Flexible array members should not be declared](https://rules.sonarsource.com/c/RSPEC-2324)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2324)

### [Preprocessor directives should not be indented](https://rules.sonarsource.com/c/RSPEC-1915)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1915)

### ["switch" statements should not be nested](https://rules.sonarsource.com/c/RSPEC-1821)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1821)

### [Cyclomatic Complexity of functions should not be too high](https://rules.sonarsource.com/c/RSPEC-1541)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1541)

### ["switch" statements should have "default" clauses](https://rules.sonarsource.com/c/RSPEC-131)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-131)

### ["if ... else if" constructs should end with "else" clauses](https://rules.sonarsource.com/c/RSPEC-126)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-126)

### ["typedef" should be used for function pointers](https://rules.sonarsource.com/c/RSPEC-1259)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1259)

### [Control structures should use curly braces](https://rules.sonarsource.com/c/RSPEC-121)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-121)

### [Expressions should not be too complex](https://rules.sonarsource.com/c/RSPEC-1067)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1067)

### [Macros used in preprocessor directives should be defined before use](https://rules.sonarsource.com/c/RSPEC-966)

[Bug](https://rules.sonarsource.com/c/RSPEC-966)

### [The number of arguments passed to a function should match the number of parameters](https://rules.sonarsource.com/c/RSPEC-930)

[Bug](https://rules.sonarsource.com/c/RSPEC-930)

### [Evaluation of the operand to the sizeof operator shall not contain side effects](https://rules.sonarsource.com/c/RSPEC-922)

[Bug](https://rules.sonarsource.com/c/RSPEC-922)

### [Bitwise operators should not be applied to signed operands](https://rules.sonarsource.com/c/RSPEC-874)

[Bug](https://rules.sonarsource.com/c/RSPEC-874)

### [Boolean operations should not have numeric operands, and vice versa](https://rules.sonarsource.com/c/RSPEC-867)

[Bug](https://rules.sonarsource.com/c/RSPEC-867)

### [Pointer conversions should be restricted to a safe subset](https://rules.sonarsource.com/c/RSPEC-856)

[Bug](https://rules.sonarsource.com/c/RSPEC-856)

### [Function pointers should not be converted to any other type](https://rules.sonarsource.com/c/RSPEC-855)

[Bug](https://rules.sonarsource.com/c/RSPEC-855)

### [Results of ~ and << operations on operands of underlying types unsigned char and unsigned short should immediately be cast to the operand's underlying type](https://rules.sonarsource.com/c/RSPEC-853)

[Bug](https://rules.sonarsource.com/c/RSPEC-853)

### [User-defined types should not be passed as variadic arguments](https://rules.sonarsource.com/c/RSPEC-5270)

[Bug](https://rules.sonarsource.com/c/RSPEC-5270)

### [The "<stdlib.h>" functions "bsearch" and "qsort" should not be used](https://rules.sonarsource.com/c/RSPEC-2393)

[Bug](https://rules.sonarsource.com/c/RSPEC-2393)

### [Floating point numbers should not be tested for equality](https://rules.sonarsource.com/c/RSPEC-1244)

[Bug](https://rules.sonarsource.com/c/RSPEC-1244)

### [There shall be at most one occurrence of the # or ## operators in a single macro definition](https://rules.sonarsource.com/c/RSPEC-967)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-967)

### [Parameters in a function prototype should be named](https://rules.sonarsource.com/c/RSPEC-926)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-926)

### ["goto" statement should not be used](https://rules.sonarsource.com/c/RSPEC-907)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-907)

### [Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression](https://rules.sonarsource.com/c/RSPEC-881)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-881)

### ["enum" values should not be used as operands to built-in operators other than [ ], =, ==, !=, unary &, and the relational operators <, <=, >, >=](https://rules.sonarsource.com/c/RSPEC-873)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-873)

### [Operands of "&&" and "||" should be primary (C) or postfix (C++) expressions](https://rules.sonarsource.com/c/RSPEC-868)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-868)

### [Limited dependence should be placed on operator precedence](https://rules.sonarsource.com/c/RSPEC-864)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-864)

### [The value of a complex expression should only be cast to a type that is narrower and of the same signedness as the underlying type of the expression](https://rules.sonarsource.com/c/RSPEC-851)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-851)

### [Braces should be used to indicate and match the structure in the non-zero initialization of arrays and structures](https://rules.sonarsource.com/c/RSPEC-835)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-835)

### [Array declarations should include an explicit size specification](https://rules.sonarsource.com/c/RSPEC-834)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-834)

### ["typedef" names should be unique identifiers](https://rules.sonarsource.com/c/RSPEC-802)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-802)

### [Identifiers should not be longer than 31 characters](https://rules.sonarsource.com/c/RSPEC-799)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-799)

### [All uses of the #pragma directive should be documented](https://rules.sonarsource.com/c/RSPEC-793)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-793)

### [Assembly language should be encapsulated and isolated](https://rules.sonarsource.com/c/RSPEC-784)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-784)

### [Functions that are not used in a project should be removed](https://rules.sonarsource.com/c/RSPEC-5536)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5536)

### [Track parsing failures](https://rules.sonarsource.com/c/RSPEC-2260)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2260)

### [Files should not be too complex](https://rules.sonarsource.com/c/RSPEC-1908)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1908)

### [The ternary operator should not be used](https://rules.sonarsource.com/c/RSPEC-1774)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1774)

### [Functions/methods should not have too many lines](https://rules.sonarsource.com/c/RSPEC-138)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-138)

### [Track uses of "NOSONAR" comments](https://rules.sonarsource.com/c/RSPEC-1291)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1291)

### ["for" loop stop conditions should be invariant](https://rules.sonarsource.com/c/RSPEC-127)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-127)

### [Statements should be on separate lines](https://rules.sonarsource.com/c/RSPEC-122)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-122)

### ["switch case" clauses should not have too many lines of code](https://rules.sonarsource.com/c/RSPEC-1151)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1151)

### [Functions should not contain too many return statements](https://rules.sonarsource.com/c/RSPEC-1142)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1142)

### [Magic numbers should not be used](https://rules.sonarsource.com/c/RSPEC-109)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-109)

### [Files should not have too many lines of code](https://rules.sonarsource.com/c/RSPEC-104)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-104)

### [Lines should not be too long](https://rules.sonarsource.com/c/RSPEC-103)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-103)

### [Function parameters' initial values should not be ignored](https://rules.sonarsource.com/c/RSPEC-1226)

[Bug](https://rules.sonarsource.com/c/RSPEC-1226)

### [Preprocessor operators "#" and "##" should not be used](https://rules.sonarsource.com/c/RSPEC-968)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-968)

### [Structure and union types should be complete at the end of a translation unit](https://rules.sonarsource.com/c/RSPEC-950)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-950)

### [Switch statement conditions should not have essentially boolean type](https://rules.sonarsource.com/c/RSPEC-920)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-920)

### ["continue" should not be used](https://rules.sonarsource.com/c/RSPEC-909)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-909)

### [Tests of non-Boolean values against zero should be explicit](https://rules.sonarsource.com/c/RSPEC-883)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-883)

### [Signed and unsigned types should not be mixed in expressions](https://rules.sonarsource.com/c/RSPEC-845)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-845)

### [typedefs that indicate size and signedness should be used in place of the basic types](https://rules.sonarsource.com/c/RSPEC-813)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-813)

### [Appropriate char types should be used for character and integer values](https://rules.sonarsource.com/c/RSPEC-810)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-810)

### [Source code should only use /\* ... \*/ style comments](https://rules.sonarsource.com/c/RSPEC-787)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-787)

### [GNU extensions should not be used](https://rules.sonarsource.com/c/RSPEC-3715)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3715)

### [Methods should not return constants](https://rules.sonarsource.com/c/RSPEC-3400)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3400)

### [Label names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-3222)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-3222)

### [Enumeration values should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-2343)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2343)

### [Enumeration names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-2342)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-2342)

### [Comment styles "//" and "/\* ... \*/" should not be mixed within a file](https://rules.sonarsource.com/c/RSPEC-1917)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1917)

### ["union" names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-1878)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1878)

### [Constants should come first in equality tests](https://rules.sonarsource.com/c/RSPEC-1772)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1772)

### [Type specifiers should be listed in a standard order](https://rules.sonarsource.com/c/RSPEC-1749)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1749)

### [Track "TODO" and "FIXME" comments that do not contain a reference to a person](https://rules.sonarsource.com/c/RSPEC-1707)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1707)

### [The prefix increment/decrement form should be used](https://rules.sonarsource.com/c/RSPEC-1705)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1705)

### ["struct" names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-1642)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1642)

### [File names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-1578)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1578)

### [Macro names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-1543)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1543)

### [Comments should not be located at the end of lines of code](https://rules.sonarsource.com/c/RSPEC-139)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-139)

### [break statements should not be used except for switch cases](https://rules.sonarsource.com/c/RSPEC-1227)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1227)

### [Local variable and function parameter names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-117)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-117)

### [Field names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-116)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-116)

### [Lines should not end with trailing whitespaces](https://rules.sonarsource.com/c/RSPEC-1131)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1131)

### [Files should contain an empty newline at the end](https://rules.sonarsource.com/c/RSPEC-113)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-113)

### [Tabulation characters should not be used](https://rules.sonarsource.com/c/RSPEC-105)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-105)

### [A function should have a single point of exit at the end of the function](https://rules.sonarsource.com/c/RSPEC-1005)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1005)

### [Function names should comply with a naming convention](https://rules.sonarsource.com/c/RSPEC-100)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-100)

### [Track comments matching a regular expression](https://rules.sonarsource.com/c/RSPEC-5639)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-5639)

### [Track instances of the "#error" preprocessor directive being reached](https://rules.sonarsource.com/c/RSPEC-1914)

[Code Smell](https://rules.sonarsource.com/c/RSPEC-1914)