C static code analysis

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

Function names should be used either as a call with a parameter list or with the "&" operator

Analyze your code

Code Smell   Critical   based-on-misra suspicious

Using a "bald" function name is likely a bug. Rather than testing the return value of a function with a `void` parameter list, it implicitly retrieves the address of that function in memory. If that's truly what's intended, then it should be made explicit with the use of the `&` (address-of) operator. If it's not, then a parameter list (even an empty one) should be added after the function name.

Noncompliant Code Example

```
int func(void) {
  // ...
}

void f2(int a, int b) {
  // ...
  if (func) {  // Noncompliant - tests that the memory addres
    //...
  }
  // ...
}
```

Compliant Solution

```
void f2(int a, int b) {
  // ...
  if (func()) {  // tests that the return value of func() > 0
    //...
  }
  // ...
}
```

Exceptions

Callback functions are a common occurrence and are usually not passed with a preceding &. There is however little ambiguity so this rule ignores function identifiers when used as a parameter of a function call.

```
void foo() {
  // ...
}

registerEvent(AnEvent, foo);
```

See

- MISRA C:2004, 16.9 - A function identifier shall only be used with either a preceding &, or with a parenthesized parameter list, which may be empty.
- MISRA C++:2008, 8-4-4 - A function identifier shall only be used to call the

function or it shall be preceded by &.

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

Stack allocated memory and non-owned memory should not be freed

Bug

Closed resources should not be accessed

Bug

Dynamically allocated memory should be released

Bug

Freed memory should not be used