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
Vulnerability

POSIX functions should not be called with arguments that trigger buffer overflows
Vulnerability

XML parsers should not be vulnerable to XXE attacks
Vulnerability

Function-like macros should not be invoked without all of their arguments
Bug

The address of an automatic object should not be assigned to another object that may persist after the first object has ceased to exist
Bug

"pthread_mutex_t" should be unlocked in the reverse order they were locked
Bug

"pthread_mutex_t" should be properly initialized and destroyed
Bug

"pthread_mutex_t" should not be consecutively locked or unlocked twice
Bug

Functions with "noreturn" attribute should not return
Bug

"memcmp" should only be called with pointers to trivially copyable types with no padding
Bug

Unused function parameters should be removed

Analyze your code

Code Smell   Major   based-on-misra cert unused

Unused parameters are misleading. Whatever the values passed to such parameters, the behavior will be the same.

There are some cases when you want to have an unused parameter (usually because the function has to conform to a fixed prototype, because it is virtual or it is going to be called from a template). In this case, and if the parameter is never used, an accepted practice is to leave it unnamed. If it is only sometimes used (for instance, depending on conditional compilation), you may, since C++17, use the `[[maybe_unused]]` attribute to be explicit about it.

```
void f([[maybe_unused]] int i) {
  assert(i < 42); // In optimized mode, this assert will be r
}
```

In case of Objective-C it is acceptable to have unused parameters if the method is supposed to be overridden.

Noncompliant Code Example

```
void doSomething(int a, int b) { // Noncompliant, "b" is unus
  compute(a);
}
```

Compliant Solution

```
void doSomething(int a) {
  compute(a);
}
```

See

- MISRA C++:2008, 0-1-11 - There shall be no unused parameters (named or unnamed) in nonvirtual functions.
- MISRA C:2012, 2.7 - There should be no unused parameters in functions
- CERT, MSC12-C. - Detect and remove code that has no effect or is never executed
- C++ Core Guidelines - F.9 - Unused parameters should be unnamed

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