

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Kubernetes
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



C++ static code analysis

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

All rules578

Vulnerability13

Bug111

Security Hotspot18

Code Smell436

Quick Fix68

Tags

Search by name...

"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

Assigning to an optional should directly target the optional

Bug

Result of the standard remove algorithms should not be ignored

Bug

"std::scoped_lock" should be created with constructor arguments

Bug

Objects should not be sliced

Bug

Immediately dangling references should not be created

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

Single-bit named bit fields should not be of a signed type

Analyze your code

BugMajorbased-on-misra

The values that can be represented by a signed bit field with a length of one bit may not meet developer expectations. For example according to the C99 Standard, Section 6.2.6.2, a single-bit signed bit-field has a single (one) sign bit and no (zero) value bits.

This rule does not apply to unnamed bit fields, as their values cannot be accessed.

Noncompliant Code Example

```
signed int f:1; // Noncompliant; there's only room here for
```

Compliant Solution

```
unsigned int f:1;
```

or

```
signed int:1; // unnamed
```

or

```
signed int f:2;
```

See

- MISRA C:2004, 6.5 - Bit fields of type signed int shall be at least 2 bits long
- MISRA C:2012, 6.2 - Single-bit named bit fields shall not be of a signed type
- MISRA C++:2008, 9-6-4 - Named bit-fields with signed integer type shall have a length of more than one bit

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

 Bug
"std::move" and "std::forward" should not be confused  Bug
A call to "wait()" on a "std::condition_variable" should have a condition  Bug
A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic_cast  Bug
Functions with "noreturn" attribute should not return  Bug
RAII objects should not be temporary  Bug
"memcmp" should only be called with pointers to trivially copyable types with no padding  Bug
"memcpy", "memmove", and "memset" should only be called with pointers to trivially copyable types  Bug
"std::auto_ptr" should not be used  Bug
Destructors should be "noexcept"  Bug