C++ static code analysis

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

All rules 578 | Vulnerability 13 | Bug 111 | Security Hotspot 18 | Code Smell 436 | Quick Fix 68

Tags

Search by name...

"reinterpret_cast" should not be used

Analyze your code

Code Smell Major cppcoreguidelines pitfall

Because `reinterpret_cast` does not perform any type safety validations, it is capable of performing dangerous conversions between unrelated types.

Since C++20, a `std::bit_cast` should be used instead of `reinterpret_cast` to reinterpret a value as being of a different type of the same length preserving its binary representation, as the behavior of `reinterpret_cast` is undefined in such case.

This rule raises an issue when `reinterpret_cast` is used.

Noncompliant Code Example

```
class A { public: virtual ~A(){} };
class B : public A { public: void doSomething(){} };

void func(A *a, float f) {
  if (B* b = reinterpret_cast<B*>(a)) { // Noncompliant
    b->doSomething();
  }
  int x = *reinterpret_cast<int*>(f); // Noncompliant
}
```

Compliant Solution

```
class A { public: virtual ~A(){} };
class B : public A { public: void doSomething(){} };

void func(A *a, float f) {
  if (B* b = dynamic_cast<B*>(a)) {
    b->doSomething();
  }
  int x = std::bit_cast<int>(f);
}
```

See

- CppCoreGuidelines, Type safety profile - Type.1: Don't use reinterpret_cast.

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