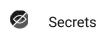
Search by name...





ABAP

Арех Арех

C C

© C++

CloudFormation

COBOL COBOL

C# C#

∃ css

X Flex

GO Go

5 HTML

🐇 Java

Js JavaScript

Kotlin

Kubernetes

6 Objective C

PHP

PL/I

PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

Text

TS TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



C++ static code analysis

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

All 578 rules

Outlinerability 13 Rug 111 Security Hotspot

Security 18 Security Hotspot

Outlinerability 13 Rug 111 Security 18 Security Hotspot

Outlinerability 19 Fix

Tags

"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

"pthread_mutex_t" should not be consecutively locked or unlocked

initialized and destroyed

📆 Bug

Code Smell		ntion
For better readal	oility, do not put more than one s	statement on a single line.
Noncompliant C	ode Example	
foo(); bar	(); // Noncompliant	
Compliant Solut	ion	
foo();		
bar(); Exceptions		
Exceptions Control flow stat if (condit:	ements with a single nested station) doSomething();	// Compliant
Exceptions Control flow state if (condit: while (condit)	ion) doSomething(); dition) doSomething();	// Compliant
Exceptions Control flow state if (condit: while (condit) case or default are ignored. switch (for case 0:	ion) doSomething(); dition) doSomething(); t statements containing a sing	<pre>// Compliant // Compliant e statement and followed by brea // Compliant</pre>
Exceptions Control flow state if (condition while (condition case or default are ignored. Switch (for case 0: default: }	<pre>ion) doSomething(); dition) doSomething(); t statements containing a sing o) { doSomething(); break;</pre>	<pre>// Compliant // Compliant e statement and followed by brea // Compliant // Compliant</pre>

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy

🖟 Bug
"std::move" and "std::forward" should not be confused
∰ Bug
A call to "wait()" on a "std::condition_variable" should have a condition
n 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
्रे 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
n Bug
Destructors should be "noexcept"
🖟 Bug