



XML



C static code analysis

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

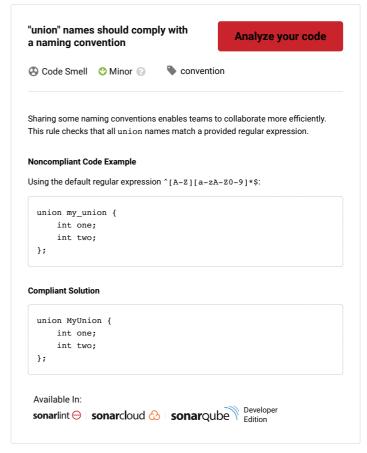
All 311 rules	6 Vulnerability 13	∰ Bug 74)	Security Hotspot	Code Smell (206)	Quick 14 Fix
---------------	--------------------	--------------------------	------------------	------------------	-----------------

Tags

"memset" sensitive	should not be used to delete data
6 Vulner	rability
	nctions should not be called ments that trigger buffer
6 Vulner	ability
XML pars to XXE att	ers should not be vulnerable acks
6 Vulner	ability
	like macros should not be rithout all of their arguments
<table-of-contents> Bug</table-of-contents>	
should no object tha	ess of an automatic object t be assigned to another it may persist after the first s ceased to exist
<table-of-contents> Bug</table-of-contents>	
. –	mutex_t" should be unlocked erse order they were locked
<table-of-contents> Bug</table-of-contents>	
	mutex_t" should be properly and destroyed
<table-of-contents> Bug</table-of-contents>	
	mutex_t" should not be vely locked or unlocked
<table-of-contents> Bug</table-of-contents>	
Functions should no	with "noreturn" attribute t return
🕕 Bug	
"memcmr	o" should only be called with

pointers to trivially copyable types with no padding

👬 Bug



Search by name.

© 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

Stack allocated memory and nonowned memory should not be freed

R
Bug

Closed resources should not be accessed

Bug

Dynamically allocated memory should be released

Bug

Freed memory should not be used