- Secrets
- SAP ABAP
- APEX Apex
- c C
- C++
- CloudFormation
- COBOL
- C# C#
- CSS
- Flex
- GO Go
- HTML
- Java
- JS JavaScript
- Kotlin
- Kubernetes
- Objective C
- php PHP
- PL/I PL/I
- PL/SQL 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 rules 311 | 🔒 Vulnerability 13 | 🐞 Bug 74 | 🛡 Security Hotspot 18 | ⊙ Code Smell 206 | ⚡ Quick Fix 14 |

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

"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

"static" should not be used for the size of an array parameter

Analyze your code

⊙ Code Smell   ⚠ Critical ⍰   🏷 based-on-misra pitfall

Theoretically, the use of the `static` keyword on the size of an array parameter means you can assume that only arrays of at least that size will be passed as arguments to the function. I.e. a function parameter of `int my_array[static 10]` means that `my_array` will always be *at least* 10 elements long. If it is not, the behavior is undefined.

In practice, the use of `static` on the size of an array parameter means the compiler might issue a warning if a noncompliant array is passed to the function - a warning that might or might not be ignored. Therefore, in practice the use of `static` on an array parameter's size merely lends a false sense of security, and `static` should not be used in this context.

Note that for some compiler/processor combinations, more efficient code can be generated when `static` is used, but these combinations are limited, and the benefit does not outweigh the cost.

Noncompliant Code Example

```
int total (int size, int my_array[static 10]) {...}
```

Compliant Solution

```
int total (int size, int my_array[10]) {...}
```

See

- MISRA C:2012, 17.6 - The declaration of an array parameter shall not contain the static keyword between the []

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