C static code analysis: Array declarations should include an explicit size specification

2 minutes

It is possible to declare an array without explicitly specifying its size, but using an explicit size declaration is clearer, and is therefore preferred.

Noncompliant Code Example

```
int arr1 [ ];  // Noncompliant; nothing specified
int arr2 [ ] = { [0] = 1, [12] = 36, [4] = 93 }; // Noncompliant; highest index determines size. May be difficult to spot
int pirate [ ] = { 2, 4, 8, 42, 501, 90210, 7, 1776 }; // Noncompliant; size is implicit, not explicit
```

Compliant Solution

```
int arr1 [10];
int arr2 [13] = { [0] = 1, [12] = 36, [4] = 93 };
int pirate [10] = { 2, 4, 8, 42, 501, 90210, 7, 1776 }; // Implicitly-assigned size was 8. Desired size was 10.
```

See

- MISRA C:2004, 8.12 - When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by

initialisation

- MISRA C++:2008, 3-1-3 - When an array is declared, its size shall either be stated explicitly or defined implicitly by initialization
- MISRA C:2012, 8.11 - When an array with external linkage is declared, its size should be explicitely specified
- MISRA C:2012, 9.5 - Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly
- CERT, ARR02-C. - Explicitly specify array bounds, even if implicitly defined by an initializer