

C static code analysis: "nonnull" pointers should not be set to null

1 minute

Functions return values and parameters values marked `nonnull` are assumed to have non-null values and are not typically null-checked before use. Therefore setting one of these values to `null`, could cause null pointer dereferences at runtime.

Noncompliant Code Example

```
__attribute__((returns_nonnull))
int* nonnull(__attribute__((nonnull)) int* parameter) {
    parameter = 0; // Noncompliant - "parameter" is marked
    "nonnull" but is set to null.
    nonnull(0); // Noncompliant - Parameter "parameter" to this call
    is marked "nonnull" but null is passed.
    return 0; // Noncompliant - This function's return value is
    marked "nonnull" but null is returned.
}
```

See

- [MITRE, CWE-476](#) - NULL Pointer Dereference
- [CERT, EXP01-J](#) - Do not use a null in a case where an object is required

