

C++ static code analysis: Appropriate arguments should be passed to stream functions

1-2 minutes

The standard C library includes a number of functions for handling streams. If not called correctly, these functions can have undefined behavior. More specifically:

- FILE* should be checked for null before being used in a function that accesses the file content
- The third argument of fseek must be SEEK_SET, SEEK_END, or SEEK_CUR

Noncompliant Code Example

```
FILE *file1 = fopen("myFile", "r");
fseek(file1, 1, SEEK_SET); // Noncompliant, file could be NULL
fclose(file1);

FILE *file2 = tmpfile();
ftell(file2); // Noncompliant, file could be NULL
if (file2) {
    fseek(file2, 1, 3); // Noncompliant, third argument should either be
                        // SEEK_SET, SEEK_CUR or SEEK_END
}
fclose(file2);
```

Compliant Solution

```
FILE *file1 = fopen("myFile", "r");
if (file1) {
    fseek(file1, 1, SEEK_SET);
    fclose(file1);
}
```

```
FILE *file2 = tmpfile();  
if (file2) {  
    ftell(file2);  
    fseek(file2, 1, SEEK_END);  
    fclose(file2);  
}
```