FIO11-C. Take care when specifying the mode parameter of fopen()

The C Standard identifies specific strings to use for the `mode` on calls to `fopen()` and `fopen_s()`. C11 provides a new mode flag, `x`, that provides the mechanism needed to determine if the file that is to be opened exists. To be strictly conforming and portable, one of the strings from the following table (adapted from the C Standard, subclause 7.21.5.2 [ISO/IEC 9899:2011]) must be used:

Strings to Use for the Mode on Calls to `fopen()` and `fopen_s()`

`mode` String	Result
`r`	Open text file for reading
`w`	Truncate to zero length or create text file for writing
`wx`	Create text file for writing
`a`	Append; open or create text file for writing at end-of-file
`rb`	Open binary file for reading
`wb`	Truncate to zero length or create binary file for writing
`wbx`	Create binary file for writing
`ab`	Append; open or create binary file for writing at end-of-file
`r+`	Open text file for update (reading and writing)
`w+`	Truncate to zero length or create text file for update
`w+x`	Create text file for update
`a+`	Append; open or create text file for update, writing at end-of-file
`r+b` or `rb+`	Open binary file for update (reading and writing)
`w+b` or `wb+`	Truncate to zero length or create binary file for update
`w+bx` or `wb+x`	Create binary file for update
`a+b` or `ab+`	Append; open or create binary file for update, writing at end-of-file

If the `mode` string begins with one of these sequences, the implementation might choose to ignore the remaining characters, or it might use them to select different kinds of files.

When calling `fopen_s()`, any of the mode strings used for writing (`w` or `a`) may be prefixed with the `u` character to give the file system default access permissions.

An implementation may define additional `mode` strings, but only the modes shown in the table are fully portable and C compliant. Beware that Microsoft Visual Studio 2012 and earlier do not support the `x` or `u` mode characters [MSDN].

Risk Assessment

Using a `mode` string that is not recognized by an implementation may cause the call to `fopen()` to fail.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
FIO11-C	Medium	Probable	Medium	**P8**	**L2**

Automated Detection

Tool	Version	Checker	Description
Compass/ROSE			
LDRA tool suite	9.7.1	**590 S**	Partially implemented
Polyspace Bug Finder	R2019b	CERT C: Rec. FIO11-C	Checks for bad file access mode or status (rec. fully covered)

Related Vulnerabilities

Search for vulnerabilities resulting from the violation of this rule on the CERT website.

Related Guidelines

SEI CERT C++ Coding Standard	VOID FIO11-CPP. Take care when specifying the mode parameter of fopen()

Bibliography

[ISO/IEC 9899:2011]	Subclause 7.21.5.3, "The `fopen` Function"

← ↑ →