ISO/IEC JTC 1/SC 22/WG 23 N 0336

Proposed changes to Clause 6 introduction

Date 4 May 2011
Contributed by Jim Moore
Original file name
Notes

The intent of this proposal is to clarify that our descriptions are based on the behaviour specified by the language standards and that other vulnerabilities may exist in non-standard implementations. The text below is copied from N0335; my proposal would make the indicated changes.

6.1 General

This clause provides language-independent descriptions of vulnerabilities in programming languages that can lead to application vulnerabilities. Each description provides:

- a summary of the vulnerability,
- characteristics of languages where the vulnerability may be found,
- typical mechanisms of failure,
- techniques that programmers can use to avoid the vulnerability, and
- ways that language designers can modify language specifications in the future to help programmers mitigate the vulnerability.

Annexes provide descriptions of how the vulnerabilities are manifested in various specific programming languages. In each case, the behaviour of the language is assumed to be as specified by the standard cited in the annex. Clearly, programs would have different vulnerabilities in a non-standard implementation. Examples of non-standard implementations include: compilers written to implement some specification other than the standard; use of non-standard vendor extensions to the language; and use of compiler switches providing alternative semantics.

6.2 Terminology

The following descriptions are written in a language-independent manner except when specific languages are used in examples. The annexes may be consulted for language specific descriptions.

~~The standard for a programming language provides definitions for that language's constructs.~~

This clause will, in general, use the terminology that is most natural to the description of each individual vulnerability. Hence terminology may differ from to description to description.