ISO/IEC JTC 1/SC 22/WG 23 N 0366

Revised proposed revision to Clause 4.3 to describe use of annexes

Date	3 October 2011
Contributed by	Jim Moore
Original file name	
Notes	Revision of N 0365

James Moore 10/3/11 1:13 PM
Deleted: 5

James Moore 10/3/11 1:14 PM
Deleted: *P*

James Moore 10/3/11 1:13 PM
Deleted: 1

James Moore 10/3/11 1:14 PM
Deleted: Responds to Action Item 18-06

The current text of 4.3 is shown below. Proposed changes are shown using Track Changes. Further changes were made during Meeting #19.

4.3 How to Use This Document

This Technical Report gathers descriptions of programming language vulnerabilities, as well as selected application vulnerabilities, which have occurred in the past and are likely to occur again. Each vulnerability and its possible mitigations are described in the body of the report in a language-independent manner. In addition, annexes for particular languages describe the vulnerabilities and their mitigations in a manner specific to the language.

Because new vulnerabilities are always being discovered, it is anticipated that this Technical Report will be revised and new descriptions added. For that reason, a scheme that is distinct from sub-clause numbering has been adopted to identify the vulnerability descriptions. Each description has been assigned an arbitrarily generated, unique three-letter code. These codes should be used in preference to sub-clause numbers when referencing descriptions because they will not change as additional descriptions are added to future editions of this Technical Report.

James Moore 10/3/11 12:50 PM
Deleted:

The main part of this Technical Report contains descriptions that are intended to be language-independent to the greatest possible extent. Annexes apply the generic guidance to particular programming languages.

This Technical Report has been written with several possible usages in mind:

- Programmers familiar with the vulnerabilities of a specific language can reference the guide for more generic descriptions and their manifestations in less familiar languages.

- Tool vendors can use the three-letter codes as a succinct way to "profile" the selection of vulnerabilities considered by their tools.

- Individual organizations may wish to write their own coding standards intended to reduce the number of vulnerabilities in their software products. The guide can assist in the selection of vulnerabilities to be addressed in those standards and the selection of coding guidelines to be enforced.

- Organizations or individuals selecting a language for use in a project may want to consider the vulnerabilities inherent in various candidate languages.

The descriptions include suggestions for ways of avoiding the vulnerabilities. Some are simply the avoidance of particular coding constructs, but others may involve increased review or other verification and validation methods. Source code checking tools can be used to automatically enforce some coding rules and standards.

Clause 2 provides Normative references, and Clause 3 provides Terms, definitions, symbols and conventions.

Clause 4 provides the basic concepts used for this Technical Report.

Clause 5, Vulnerability Issues, provides rationale for this Technical Report and explains how many of the vulnerabilities occur.

Clause 6, Programming Language Vulnerabilities, 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.

Clause 7, Application Vulnerabilities, provides descriptions of selected application vulnerabilities which have been found and exploited in a number of applications and which have well known mitigation techniques, and which result from design decisions made by coders in the absence of suitable language library routines or other mechanisms. For these vulnerabilities, each description provides:

- a summary of the vulnerability,
- typical mechanisms of failure, and
- techniques that programmers can use to avoid the vulnerability.

Annex A, Vulnerability Taxonomy and List, is a categorization of the vulnerabilities of this report in the form of a hierarchical outline and a list of the vulnerabilities arranged in alphabetic order by their three letter code.

James Moore 10/1/11 8:16 AM
Deleted: D

James Moore 10/1/11 8:20 AM
Deleted: Outline

Annex B, Language Specific Vulnerability Template, is a template for the writing of programming language specific annexes that explain how the vulnerabilities are realized in that programming language (or show how they are absent), and how they might be mitigated in language-specific terms.

James Moore 10/1/11 8:20 AM
Deleted: E

James Moore 10/3/11 1:00 PM
Deleted: from clause 6

Additional annexes, each named for a particular programming language, list the vulnerabilities and describe whether each vulnerability appears in the specific language and, if present, how it may be mitigated in that language. All of the language-dependent descriptions assume that the user adheres to the standard for the language as listed in the first sub-section of each annex.

James Moore 10/1/11 8:21 AM
Deleted: Future revisions of this Technical Report are planned to contain language-specific annexes that are developed using Annex E