

# ISO/IEC JTC 1/SC 22/WG 23 N 0323

*Proposal for alignment of numbering between Clauses and Annexes*

**Date** 24 March 2011  
**Contributed by** Editor and secretary  
**Original file name**  
**Notes**

## Proposed alignment of Clause 6 and Annexes

<b>Clause 6 Programming Language Vulnerabilities</b>	<b>C. Vulnerability descriptions for the language C</b>
<p><b>6.1 General</b> This clause provides language-independent descriptions of vulnerabilities in programming languages that can lead to application vulnerabilities. Each description provides:</p> <ul style="list-style-type: none"><li>• a summary of the vulnerability,</li><li>• characteristics of languages where the vulnerability may be found,</li><li>• typical mechanisms of failure,</li><li>• techniques that programmers can use to avoid the vulnerability, and</li><li>• ways that language designers can modify language specifications in the future to help programmers mitigate the vulnerability.</li></ul>	<p><b>C.1 Identification of standards and associated documents</b> ...</p>
<p><b>6.2 Terminology</b> 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 description to description.</p>	<p><b>C.2 General terminology and concepts</b> ...</p>

6.3 Type System [IHN]	C.3 Type System [IHN]
-----------------------	-----------------------

## Proposed alignment of Clause 6 and Clause 7

Clause 6 Programming Language Vulnerabilities	Clause 7 Application Vulnerabilities
<p><b>6.1 General</b></p> <p>This clause provides language-independent descriptions of vulnerabilities in programming languages that can lead to application vulnerabilities. Each description provides:</p> <ul style="list-style-type: none"> <li>• a summary of the vulnerability,</li> <li>• characteristics of languages where the vulnerability may be found,</li> <li>• typical mechanisms of failure,</li> <li>• techniques that programmers can use to avoid the vulnerability, and</li> <li>• ways that language designers can modify language specifications in the future to help programmers mitigate the vulnerability.</li> </ul>	<p><b>7.1 General</b></p> <p>This clause 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:</p> <ul style="list-style-type: none"> <li>• a summary of the vulnerability,</li> <li>• typical mechanisms of failure, and</li> <li>• techniques that programmers can use to avoid the vulnerability.</li> </ul>
<p><b>6.2 Terminology</b></p> <p>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.</p> <p>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 description to description.</p>	<p><b>7.2 Terminology</b></p> <p>These vulnerabilities are application-related rather than language-related. They are written in a language-independent manner, and there are no corresponding sections in the annexes.</p>
6.3 Type System [IHN]	7.3 Unspecified Functionality [BVQ]