
**Information technology — Programming
languages — Guidance to avoiding
vulnerabilities in programming languages
through language selection and use**

*Technologies de l'information — Langages de programmation —
Conduite pour éviter les vulnérabilités dans les langages de
programmation à travers la sélection et l'usage de la langue*

Withhold

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Withdrawn



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions, symbols and conventions	1
3.1 Terms and definitions, symbols and conventions.....	1
3.2 Symbols and conventions.....	3
4 Basic Concepts	4
4.1 Not in Scope.....	4
4.2 Approach	4
4.3 Intended Audience.....	4
4.4 How to Use This Document	5
5 Vulnerability issues	8
5.1 Issues arising from incomplete or evolving language specifications	8
5.2 Issues arising from human cognitive limitations.....	11
5.3 Issues arising from a lack of predictable execution	12
5.4 Issues arising from the lack of portability and interoperability	12
5.5 Issues arising from inadequate language intrinsic support.....	13
5.6 Issues arising from language features prone to erroneous use.....	13
6 Programming Language Vulnerabilities	14
6.1 General	14
6.2 Obscure Language Features [BRS]	14
6.3 Unspecified Behaviour [BOF].....	15
6.4 Undefined Behaviour [EWF]	17
6.5 Implementation-defined Behaviour [FAB]	18
6.6 Deprecated Language Features [MEM].....	20
6.7 Pre-processor Directives [NMP].....	21
6.8 Choice of Clear Names [NAI].....	23
6.9 Choice of Filenames and other External Identifiers [AJN].....	25
6.10 Unused Variable [XYR]	26
6.11 Identifier Name Reuse [YOW]	27
6.12 Namespace Issues [BJL].....	30
6.13 Type System [IHN].....	31
6.14 Bit Representations [STR].....	34
6.15 Floating-point Arithmetic [PLF]	35
6.16 Enumerator Issues [CCB]	38
6.17 Numeric Conversion Errors [FLC]	40

6.18	String Termination [CJM]	42
6.19	Boundary Beginning Violation [XYX]	43
6.20	Unchecked Array Indexing [XYZ]	44
6.21	Unchecked Array Copying [XYW]	46
6.22	Buffer Overflow [XZB]	47
6.23	Pointer Casting and Pointer Type Changes [HFC]	49
6.24	Pointer Arithmetic [RVG]	50
6.25	Null Pointer Dereference [XYH]	51
6.26	Dangling Reference to Heap [XYK]	52
6.27	Templates and Generics [SYM]	54
6.28	Inheritance [RIP]	56
6.29	Initialization of Variables [LAV]	57
6.30	Wrap-around Error [XYY]	59
6.31	Sign Extension Error [XZI]	60
6.32	Operator Precedence/Order of Evaluation [JCW]	61
6.33	Side-effects and Order of Evaluation [SAM]	63
6.34	Likely Incorrect Expression [KOA]	64
6.35	Dead and Deactivated Code [XYQ]	66
6.36	Switch Statements and Static Analysis [CLL]	68
6.37	Demarcation of Control Flow [EOJ]	69
6.38	Loop Control Variables [TEX]	70
6.39	Off-by-one Error [XZH]	71
6.40	Structured Programming [EWD]	73
6.41	Passing Parameters and Return Values [CSJ]	74
6.42	Dangling References to Stack Frames [DCM]	77
6.43	Subprogram Signature Mismatch [OTR]	79
6.44	Recursion [GDL]	80
6.45	Returning Error Status [NZN]	81
6.46	Termination Strategy [REU]	84
6.47	Extra Intrinsics [LRM]	85
6.48	Type-breaking Reinterpretation of Data [AMV]	87
6.49	Memory Leak [XYL]	89
6.50	Argument Passing to Library Functions [TRJ]	90
6.51	Dynamically-linked Code and Self-modifying Code [NYY]	91
6.52	Library Signature [NSQ]	92
6.53	Unanticipated Exceptions from Library Routines [HJW]	93
7	Application Vulnerabilities	95
7.1	Adherence to Least Privilege [XYN]	95
7.2	Privilege Sandbox Issues [XYO]	95
7.3	Executing or Loading Untrusted Code [XYS]	97
7.4	Unspecified Functionality [BVQ]	98
7.5	Distinguished Values in Data Types [KLK]	99
7.6	Memory Locking [XZX]	100

7.7	Resource Exhaustion [XZP]	101
7.8	Injection [RST]	102
7.9	Cross-site Scripting [XYT]	105
7.10	Unquoted Search Path or Element [XZQ]	108
7.11	Improperly Verified Signature [XZR]	108
7.12	Discrepancy Information Leak [XZL]	109
7.13	Sensitive Information Uncleared Before Release [XZK]	110
7.14	Path Traversal [EWR]	111
7.15	Missing Required Cryptographic Step [XZS]	113
7.16	Insufficiently Protected Credentials [XYM]	113
7.17	Missing or Inconsistent Access Control [XZN]	114
7.18	Authentication Logic Error [XZO]	115
7.19	Hard-coded Password [XYP]	117
	Annex A (informative) Guideline Selection Process	118
A.1	Selection Process	118
A.2	Cost/Benefit Analysis	118
A.3	Documenting of the selection process	119
	Annex B (informative) Template for use in proposing programming language vulnerabilities	120
B.1	6.<x> <short title> [<unique immutable identifier>]	120
	Annex C (informative) Template for use in proposing application vulnerabilities	122
C.1	7.<x> <short title> [<unique immutable identifier>]	122
	Annex D (informative) Vulnerability Outline and List	123
D.1	Vulnerability Outline	123
D.2	Vulnerability List	125
	Annex E (informative) Language Specific Vulnerability Template	127
E.1	<language>.1 Identification of standards	127
E.2	<language>.2 General terminology and concepts	127
E.3	<language>.<x> <Vulnerability Name> [<3 letter tag>]	127
	Bibliography	129

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 24772 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

All programming languages contain constructs that are incompletely specified, exhibit undefined behaviour, are implementation-dependent, or are difficult to use correctly. The use of those constructs may therefore give rise to *vulnerabilities*, as a result of which, software programs can execute differently than intended by the writer. In some cases, these vulnerabilities can compromise the safety of a system or be exploited by attackers to compromise the security or privacy of a system.

This Technical Report is intended to provide guidance spanning multiple programming languages, so that application developers will be better able to avoid the programming constructs that lead to vulnerabilities in software written in their chosen language and their attendant consequences. This guidance can also be used by developers to select source code evaluation tools that can discover and eliminate some constructs that could lead to vulnerabilities in their software or to select a programming language that avoids anticipated problems.

It should be noted that this Technical Report is inherently incomplete. It is not possible to provide a complete list of programming language vulnerabilities because new weaknesses are discovered continually. Any such report can only describe those that have been found, characterized, and determined to have sufficient probability and consequence.

Furthermore, to focus its limited resources, the working group developing this report decided to defer comprehensive treatment of several subject areas until future editions of the report. These subject areas include:

- Object-oriented language features (Although some simple issues related to inheritance are described in RIP)
- Concurrency
- Numerical analysis (although some simple items regarding the use of floating point are described in PLF)
- Scripting languages
- Inter-language operability
- Language-specific annexes

Information technology — Programming languages — Guidance to avoiding vulnerabilities in programming languages through language selection and use

1 Scope

This Technical Report specifies software programming language vulnerabilities to be avoided in the development of systems where assured behaviour is required for security, safety, mission critical and business critical software. In general, this guidance is applicable to the software developed, reviewed, or maintained for any application.

Vulnerabilities are described in a generic manner that is applicable to a broad range of programming languages.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 80000-2:2009, *Quantities and units — Part 2: Mathematical signs and symbols to be use in the natural sciences and technology*

ISO/IEC 2382-1:1993, *Information technology — Vocabulary — Part 1: Fundamental terms*