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

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



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested 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	xv
Introduction	xvi
1. Scope.....	1
2. Normative references	1
3. Terms and definitions, symbols and conventions	1
3.1 Terms and definitions	1
3.2 Symbols and conventions	5
4. Basic concepts	6
4.1 Purpose of this Technical Report.....	6
4.2 Intended audience.....	6
4.3 How to use this document	7
5 Vulnerability issues.....	8
5.1 Predictable execution	8
5.2 Sources of unpredictability in language specification	9
5.2.1 Incomplete or evolving specification.....	9
5.2.2 Undefined behaviour	10
5.2.3 Unspecified behaviour	10
5.2.4 Implementation-defined behaviour	10
5.2.5 Difficult features.....	10
5.2.6 Inadequate language support	10
5.3 Sources of unpredictability in language usage.....	10
5.3.1 Porting and interoperation	10
5.3.2 Compiler selection and usage	11
6. Programming Language Vulnerabilities.....	11
6.1 General.....	11
6.2 Terminology	11
6.3 Type System [IHN]	12
6.4 Bit Representations [STR]	14
6.5 Floating-point Arithmetic [PLF]	16
6.6 Enumerator Issues [CCB].....	18
6.7 Numeric Conversion Errors [FLC].....	20
6.8 String Termination [CJM]	22
6.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	23
6.10 Unchecked Array Indexing [XYZ]	25
6.11 Unchecked Array Copying [XYW]	27
6.12 Pointer Casting and Pointer Type Changes [HFC]	28
6.13 Pointer Arithmetic [RVG]	29

6.14 Null Pointer Dereference [XYH]	30
6.15 Dangling Reference to Heap [XYK]	31
6.16 Arithmetic Wrap-around Error [FIF]	34
6.17 Using Shift Operations for Multiplication and Division [PIK]	35
6.18 Sign Extension Error [XZI]	36
6.19 Choice of Clear Names [NAI]	37
6.20 Dead Store [WXQ]	39
6.21 Unused Variable [YZS]	40
6.22 Identifier Name Reuse [YOW]	41
6.23 Namespace Issues [BJL]	43
6.24 Initialization of Variables [LAV]	45
6.25 Operator Precedence/Order of Evaluation [JCW]	47
6.26 Side-effects and Order of Evaluation [SAM]	49
6.27 Likely Incorrect Expression [KOA]	50
6.28 Dead and Deactivated Code [XYQ]	52
6.29 Switch Statements and Static Analysis [CLL]	54
6.30 Demarcation of Control Flow [EOJ]	56
6.31 Loop Control Variables [TEX]	57
6.32 Off-by-one Error [XZH]	58
6.33 Structured Programming [EWD]	60
6.34 Passing Parameters and Return Values [CSJ]	61
6.35 Dangling References to Stack Frames [DCM]	63
6.36 Subprogram Signature Mismatch [OTR]	65
6.37 Recursion [GDL]	67
6.38 Ignored Error Status and Unhandled Exceptions [OYB]	68
6.39 Termination Strategy [REU]	70
6.40 Type-breaking Reinterpretation of Data [AMV]	72
6.41 Memory Leak [XYL]	74
6.42 Templates and Generics [SYM]	76
6.43 Inheritance [RIP]	78
6.44 Extra Intrinsics [LRM]	79
6.45 Argument Passing to Library Functions [TRJ]	80
6.46 Inter-language Calling [DJS]	81
6.47 Dynamically-linked Code and Self-modifying Code [NYY]	83
6.48 Library Signature [NSQ]	84
6.49 Unanticipated Exceptions from Library Routines [HJW]	86
6.50 Pre-processor Directives [NMP]	87
6.51 Suppression of Language-defined Run-time Checking [MXB]	89
6.52 Provision of Inherently Unsafe Operations [SKL]	90
6.53 Obscure Language Features [BRS]	91
6.54 Unspecified Behaviour [BQF]	92
6.55 Undefined Behaviour [EWF]	94
6.56 Implementation-defined Behaviour [FAB]	95
6.57 Deprecated Language Features [MEM]	97

7. Application Vulnerabilities.....	98
7.1 General.....	98
7.2 Terminology	99
7.3 Unspecified Functionality [BVQ]	99
7.4 Distinguished Values in Data Types [KLK]	100
7.5 Adherence to Least Privilege [XYN]	101
7.6 Privilege Sandbox Issues [XYO]	102
7.7 Executing or Loading Untrusted Code [XYS]	103
7.8 Memory Locking [XZX]	104
7.9 Resource Exhaustion [XZP]	105
7.10 Unrestricted File Upload [CBF]	107
7.11 Resource Names [HTS]	108
7.12 Injection [RST]	109
7.13 Cross-site Scripting [XYT]	112
7.14 Unquoted Search Path or Element [XZQ]	115
7.15 Improperly Verified Signature [XZR].....	115
7.16 Discrepancy Information Leak [XZL]	116
7.17 Sensitive Information Uncleared Before Use [XZK].....	117
7.18 Path Traversal [EWR]	118
7.19 Missing Required Cryptographic Step [XZS]	120
7.20 Insufficiently Protected Credentials [XYM].....	121
7.21 Missing or Inconsistent Access Control [XZN]	122
7.22 Authentication Logic Error [XZO].....	122
7.23 Hard-coded Password [XYP].....	124
8. New Vulnerabilities	125
8.1 General.....	125
8.2 Terminology	125
8.3 Concurrency – Activation [CGA]	125
8.4 Concurrency – Directed termination [CGT].....	127
8.5 Concurrent Data Access [CGX]	129
8.6 Concurrency – Premature Termination [CGS]	130
8.7 Protocol Lock Errors [CGM]	132
8.8 Inadequately Secure Communication of Shared Resources [CGY].....	134
Annex A (informative) Vulnerability Taxonomy and List	136
A.1 General	136
A.2 Outline of Programming Language Vulnerabilities	136
A.3 Outline of Application Vulnerabilities	138
A.4 Vulnerability List	138
Annex B (informative) Language Specific Vulnerability Template.....	141
Annex C (informative) Vulnerability descriptions for the language Ada	143
C.1 Identification of standards and associated documentation	143
C.2 General terminology and concepts	143

C.3 Type System [IHN].....	149
C.4 Bit Representation [STR]	149
C.5 Floating-point Arithmetic [PLF]	150
C.6 Enumerator Issues [CCB]	150
C.7 Numeric Conversion Errors [FLC]	151
C.8 String Termination [CJM]	151
C.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	152
C.10 Unchecked Array Indexing [XYZ].....	152
C.11 Unchecked Array Copying [XYW].....	152
C.12 Pointer Casting and Pointer Type Changes [HFC]	152
C.13 Pointer Arithmetic [RVG].....	153
C.14 Null Pointer Dereference [XYH]	153
C.15 Dangling Reference to Heap [XYK].....	153
C.16 Arithmetic Wrap-around Error [FIF].....	153
C.17 Using Shift Operations for Multiplication and Division [PIK]	154
C.18 Sign Extension Error [XZI]	154
C.19 Choice of Clear Names [NAI].....	154
C.20 Dead store [WXQ]	155
C.21 Unused Variable [YZS]	155
C.22 Identifier Name Reuse [YOW]	156
C.23 Namespace Issues [BJL]	156
C.24 Initialization of Variables [LAV]	156
C.25 Operator Precedence/Order of Evaluation [JCW]	157
C.26 Side-effects and Order of Evaluation [SAM].....	157
C.27 Likely Incorrect Expression [KOA]	158
C.28 Dead and Deactivated Code [XYQ]	159
C.29 Switch Statements and Static Analysis [CLL]	159
C.30 Demarcation of Control Flow [EOJ].....	160
C.31 Loop Control Variables [TEX]	160
C.32 Off-by-one Error [XZH]	160
C.33 Structured Programming [EWD]	161
C.34 Passing Parameters and Return Values [CSJ]	161
C.35 Dangling References to Stack Frames [DCM]	162
C.36 Subprogram Signature Mismatch [OTR]	162
C.37 Recursion [GDL]	163
C.38 Ignored Error Status and Unhandled Exceptions [OYB]	163
C.39 Termination Strategy [REU].....	164
C.40 Type-breaking Reinterpretation of Data [AMV]	164
C.41 Memory Leak [XYL]	165
C.42 Templates and Generics [SYM]	165
C.43 Inheritance [RIP]	166
C.44 Extra Intrinsic [LRM]	166
C.45 Argument Passing to Library Functions [TRJ]	166
C.46 Inter-language Calling [DJS].....	167

C.47 Dynamically-linked Code and Self-modifying Code [NYY]	167
C.48 Library Signature [NSQ]	167
C.49 Unanticipated Exceptions from Library Routines [HJW]	167
C.50 Pre-Processor Directives [NMP]	168
C.51 Suppression of Language-defined Run-time Checking [MXB]	168
C.52 Provision of Inherently Unsafe Operations [SKL]	168
C.53 Obscure Language Features [BRS]	169
C.54 Unspecified Behaviour [BQF]	169
C.55 Undefined Behaviour [EWF]	170
C.56 Implementation-Defined Behaviour [FAB]	171
C.57 Deprecated Language Features [MEM]	172
C.58 Implications for standardization	172
Annex D (informative) Vulnerability descriptions for the language C	174
D.1 Identification of standards and associated documents	174
D.2 General terminology and concepts	174
D.3 Type System [IHN]	177
D.4 Bit Representations [STR]	178
D.5 Floating-point Arithmetic [PLF]	179
D.6 Enumerator Issues [CCB]	180
D.7 Numeric Conversion Errors [FLC]	181
D.8 String Termination [CJM]	183
D.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	183
D.10 Unchecked Array Indexing [XYZ]	185
D.11 Unchecked Array Copying [XYW]	185
D.12 Pointer Casting and Pointer Type Changes [HFC]	186
D.13 Pointer Arithmetic [RVG]	186
D.14 Null Pointer Dereference [XYH]	187
D.15 Dangling Reference to Heap [XYK]	187
D.16 Arithmetic Wrap-around Error [FIF]	189
D.17 Using Shift Operations for Multiplication and Division [PIK]	190
D.18 Sign Extension Error [XZI]	190
D.19 Choice of Clear Names [NAI]	190
D.20 Dead Store [WXQ]	191
D.21 Unused Variable [YZS]	191
D.22 Identifier Name Reuse [YOW]	191
D.23 Namespace Issues [BJL]	192
D.24 Initialization of Variables [LAV]	192
D.25 Operator Precedence/Order of Evaluation [JCW]	193
D.26 Side-effects and Order of Evaluation [SAM]	193
D.27 Likely Incorrect Expression [KOA]	194
D.28 Dead and Deactivated Code [XYQ]	195
D.29 Switch Statements and Static Analysis [CLL]	196
D.30 Demarcation of Control Flow [EOJ]	197

D.31 Loop Control Variables [TEX].....	198
D.32 Off-by-one Error [XZH]	199
D.33 Structured Programming [EWD].....	199
D.34 Passing Parameters and Return Values [CSJ]	200
D.35 Dangling References to Stack Frames [DCM]	201
D.36 Subprogram Signature Mismatch [OTR]	201
D.37 Recursion [GDL]	202
D.38 Ignored Error Status and Unhandled Exceptions [OYB].....	202
D.39 Termination Strategy [REU]	203
D.40 Type-breaking Reinterpretation of Data [AMV].....	203
D.41 Memory Leak [XYL]	204
D.42 Templates and Generics [SYM]	204
D.43 Inheritance [RIP]	204
D.44 Extra Intrinsics [LRM]	204
D.45 Argument Passing to Library Functions [TRJ]	205
D.46 Inter-language Calling [DJS]	205
D.47 Dynamically-linked Code and Self-modifying Code [NYY]	205
D.48 Library Signature [NSQ]	206
D.49 Unanticipated Exceptions from Library Routines [HJW]	206
D.50 Pre-processor Directives [NMP]	207
D.51 Suppression of Language-defined Run-time Checking [MXB].....	208
D.52 Provision of Inherently Unsafe Operations [SKL]	208
D.53 Obscure Language Features [BRS]	208
D.54 Unspecified Behaviour [BQF]	209
D.55 Undefined Behaviour [EWF].....	209
D.56 Implementation-defined Behaviour [FAB].....	210
D.57 Deprecated Language Features [MEM]	210
D.58 Implications for standardization	211
 Annex E (<i>informative</i>) Vulnerability descriptions for the language Python	 214
E.1 Identification of standards and associated documents	214
E.2 General Terminology and Concepts	215
E.3 Type System [IHN].....	219
E.4 Bit Representations [STR]	221
E.5 Floating-point Arithmetic [PLF].....	222
E.6 Enumerator Issues [CCB]	222
E.7 Numeric Conversion Errors [FLC]	223
E.8 String Termination [CJM].....	224
E.9 Buffer Boundary Violation [HCB]	224
E.10 Unchecked Array Indexing [XYZ].....	224
E.11 Unchecked Array Copying [XYW]	224
E.12 Pointer Casting and Pointer Type Changes [HFC]	224
E.13 Pointer Arithmetic [RVG].....	224
E.14 Null Pointer Dereference [XYH]	224

E.15 Dangling Reference to Heap [XYK]	224
E.16 Arithmetic Wrap-around Error [FIF]	225
E.17 Using Shift Operations for Multiplication and Division [PIK]	225
E.18 Sign Extension Error [XZI]	225
E.19 Choice of Clear Names [NAI]	225
E.20 Dead Store [WXQ]	227
E.21 Unused Variable [YZS]	228
E.22 Identifier Name Reuse [YOW]	228
E.23 Namespace Issues [BJL]	230
E.24 Initialization of Variables [LAV]	233
E.25 Operator Precedence/Order of Evaluation [JCW]	233
E.26 Side-effects and Order of Evaluation [SAM]	234
E.27 Likely Incorrect Expression [KOA]	235
E.28 Dead and Deactivated Code [XYQ]	236
E.29 Switch Statements and Static Analysis [CLL]	237
E.30 Demarcation of Control Flow [EOJ]	237
E.31 Loop Control Variables [TEX]	238
E.32 Off-by-one Error [XZH]	239
E.33 Structured Programming [EWD]	239
E.34 Passing Parameters and Return Values [CSJ]	240
E.35 Dangling References to Stack Frames [DCM]	242
E.36 Subprogram Signature Mismatch [OTR]	242
E.37 Recursion [GDL]	242
E.38 Ignored Error Status and Unhandled Exceptions [OYB]	242
E.39 Termination Strategy [REU]	243
E.40 Type-breaking Reinterpretation of Data [AMV]	243
E.41 Memory Leak [XYL]	243
E.42 Templates and Generics [SYM]	244
E.43 Inheritance [RIP]	244
E.44 Extra Intrinsics [LRM]	244
E.45 Argument Passing to Library Functions [TRJ]	245
E.46 Inter-language Calling [DJS]	245
E.47 Dynamically-linked Code and Self-modifying Code [NYY]	246
E.48 Library Signature [NSQ]	246
E.49 Unanticipated Exceptions from Library Routines [HJW]	247
E.50 Pre-processor Directives [NMP]	247
E.51 Suppression of Language-defined Run-time Checking [MXB]	247
E.52 Provision of Inherently Unsafe Operations [SKL]	247
E.53 Obscure Language Features [BRS]	248
E.54 Unspecified Behaviour [BQF]	250
E.55 Undefined Behaviour [EWF]	251
E.56 Implementation-defined Behaviour [FAB]	252
E.57 Deprecated Language Features [MEM]	253

Annex F (informative) Vulnerability descriptions for the language Ruby	254
F.1 Identification of standards and associated documents	254
F.2 General Terminology and Concepts	254
F.3 Type System [IHN]	255
F.4 Bit Representations [STR]	256
F.5 Floating-point Arithmetic [PLF]	257
F.6 Enumerator Issues [CCB]	257
F.7 Numeric Conversion Errors [FLC]	258
F.8 String Termination [CJM]	258
F.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	258
F.10 Unchecked Array Indexing [XYZ]	258
F.11 Unchecked Array Copying [XYW]	258
F.12 Pointer Casting and Pointer Type Changes [HFC]	258
F.13 Pointer Arithmetic [RVG]	259
F.14 Null Pointer Dereference [XYH]	259
F.15 Dangling Reference to Heap [XYK]	259
F.16 Arithmetic Wrap-around Error [FIF]	259
F.17 Using Shift Operations for Multiplication and Division [PIK]	259
F.18 Sign Extension Error [XZI]	259
F.19 Choice of Clear Names [NAI]	259
F.20 Dead Store [WXQ]	260
F.21 Unused Variable [YZS]	260
F.22 Identifier Name Reuse [YOW]	260
F.23 Namespace Issues [BJL]	261
F.24 Initialization of Variables [LAV]	261
F.25 Operator Precedence/Order of Evaluation [JCW]	261
F.26 Side-effects and Order of Evaluation [SAM]	262
F.27 Likely Incorrect Expression [KOA]	263
F.28 Dead and Deactivated Code [XYQ]	263
F.29 Switch Statements and Static Analysis [CLL]	264
F.30 Demarcation of Control Flow [EOJ]	264
F.31 Loop Control Variables [TEX]	264
F.32 Off-by-one Error [XZH]	264
F.33 Structured Programming [EWD]	265
F.34 Passing Parameters and Return Values [CSJ]	265
F.35 Dangling References to Stack Frames [DCM]	266
F.36 Subprogram Signature Mismatch [OTR]	266
F.37 Recursion [GDL]	267
F.38 Ignored Error Status and Unhandled Exceptions [OYB]	267
F.39 Termination Strategy [REU]	267
F.40 Type-breaking Reinterpretation of Data [AMV]	267
F.41 Memory Leak [XYL]	267
F.42 Templates and Generics [SYM]	268
F.43 Inheritance [RIP]	268

F.44 Extra Intrinsic [LRM]	268
F.45 Argument Passing to Library Functions [TRJ]	268
F.46 Inter-language Calling [DJS]	268
F.47 Dynamically-linked Code and Self-modifying Code [NYY]	269
F.48 Library Signature [NSQ]	269
F.49 Unanticipated Exceptions from Library Routines [HJW]	269
F.50 Pre-processor Directives [NMP]	269
F.51 Suppression of Language-defined Run-time Checking [MXB]	270
F.52 Provision of Inherently Unsafe Operations [SKL]	270
F.53 Obscure Language Features [BRS]	270
F.54 Unspecified Behaviour [BQF]	270
F.55 Undefined Behaviour [EWF]	270
F.56 Implementation-defined Behaviour [FAB]	271
F.57 Deprecated Language Features [MEM]	271
Annex G (<i>informative</i>) Vulnerability descriptions for the language SPARK	272
G.1 Identification of standards and associated documentation	272
G.2 General terminology and concepts	272
G.3 Type System [IHN]	273
G.4 Bit Representation [STR]	274
G.5 Floating-point Arithmetic [PLF]	274
G.6 Enumerator Issues [CCB]	274
G.7 Numeric Conversion Errors [FLC]	274
G.8 String Termination [CJM]	274
G.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	274
G.10 Unchecked Array Indexing [XYZ]	274
G.11 Unchecked Array Copying [XYW]	274
G.12 Pointer Casting and Pointer Type Changes [HFC]	275
G.13 Pointer Arithmetic [RVG]	275
G.14 Null Pointer Dereference [XYH]	275
G.15 Dangling Reference to Heap [XYK]	275
G.16 Arithmetic Wrap-around Error [FIF]	275
G.17 Using Shift Operations for Multiplication and Division [PIK]	275
G.18 Sign Extension Error [XZI]	275
G.19 Choice of Clear Names [NAI]	275
G.20 Dead store [WXQ]	275
G.21 Unused Variable [YZS]	276
G.22 Identifier Name Reuse [YOW]	276
G.23 Namespace Issues [BJL]	276
G.24 Initialization of Variables [LAV]	276
G.25 Operator Precedence/Order of Evaluation [JCW]	276
G.26 Side-effects and Order of Evaluation [SAM]	276
G.27 Likely Incorrect Expression [KOA]	276
G.28 Dead and Deactivated Code [XYQ]	276

G.29 Switch Statements and Static Analysis [CLL]	277
G.30 Demarcation of Control Flow [EOJ]	277
G.31 Loop Control Variables [TEX]	277
G.32 Off-by-one Error [XZH]	277
G.33 Structured Programming [EWD]	277
G.34 Passing Parameters and Return Values [CSJ]	277
G.35 Dangling References to Stack Frames [DCM]	278
G.36 Subprogram Signature Mismatch [OTR]	278
G.37 Recursion [GDL]	278
G.38 Ignored Error Status and Unhandled Exceptions [OYB]	278
G.39 Termination Strategy [REU]	278
G.40 Type-breaking Reinterpretation of Data [AMV]	279
G.41 Memory Leak [XYL]	279
G.42 Templates and Generics [SYM]	279
G.43 Inheritance [RIP]	279
G.44 Extra Intrinsics [LRM]	279
G.45 Argument Passing to Library Functions [TRJ]	279
G.46 Inter-language Calling [DJS]	279
G.47 Dynamically-linked Code and Self-modifying Code [NYY]	280
G.48 Library Signature [NSQ]	280
G.49 Unanticipated Exceptions from Library Routines [HJW]	280
G.50 Pre-Processor Directives [NMP]	280
G.51 Suppression of Language-defined Run-time Checking [MXB]	280
G.52 Provision of Inherently Unsafe Operations [SKL]	280
G.53 Obscure Language Features [BRS]	280
G.54 Unspecified Behaviour [BQF]	281
G.55 Undefined Behaviour [EWF]	281
G.56 Implementation-Defined Behaviour [FAB]	281
G.57 Deprecated Language Features [MEM]	281
G.58 Implications for standardization	281
Annex H (<i>informative</i>) Vulnerability descriptions for the language PHP	282
H.1 Identification of standards and associated documentation	282
H.2 General Terminology and Concepts	283
H.3 Type System [IHN]	284
H.4 Bit Representations [STR]	285
H.5 Floating-point Arithmetic [PLF]	286
H.6 Enumerator Issues [CCB]	286
H.7 Numeric Conversion Errors [FLC]	287
H.8 String Termination [CJM]	288
H.9 Buffer Boundary Violation (Buffer Overflow) [HCB]	289
H.10 Unchecked Array Indexing [XYZ]	289
H.11 Unchecked Array Copying [XYW]	289
H.12 Pointer Casting and Pointer Type Changes [HFC]	289

H.13 Pointer Arithmetic [RVG].....	289
H.14 Null Pointer Dereference [XYH]	290
H.15 Dangling Reference to Heap [XYK]	290
H.16 Arithmetic Wrap-around Error [FIF]	290
H.17 Using Shift Operations for Multiplication and Division [PIK]	291
H.18 Sign Extension Error [XZI]	292
H.19 Choice of Clear Names [NAI]	292
H.20 Dead Store [WXQ]	293
H.21 Unused Variable [YZS]	294
H.22 Identifier Name Reuse [YOW]	294
H.23 Namespace Issues [BJL]	295
H.24 Initialization of Variables [LAV]	296
H.25 Operator Precedence/Order of Evaluation [JCW]	296
H.26 Side-effects and Order of Evaluation [SAM]	297
H.27 Likely Incorrect Expression [KOA]	298
H.28 Dead and Deactivated Code [XYQ]	299
H.29 Switch Statements and Static Analysis [CLL]	300
H.30 Demarcation of Control Flow [EOI]	300
H.31 Loop Control Variables [TEX]	301
H.32 Off-by-one Error [XZH]	301
H.33 Structured Programming [EWD]	302
H.34 Passing Parameters and Return Values [CSJ]	303
H.35 Dangling References to Stack Frames [DCM]	303
H.36 Subprogram Signature Mismatch [OTR]	303
H.37 Recursion [GDL]	304
H.38 Ignored Error Status and Unhandled Exceptions [OYB]	304
H.39 Termination Strategy [REU]	305
H.40 Type-breaking Reinterpretation of Data [AMV]	306
H.41 Memory Leak [XYL]	306
H.42 Templates and Generics [SYM]	306
H.43 Inheritance [RIP]	307
H.44 Extra Intrinsics [LRM]	307
H.45 Argument Passing to Library Functions [TRJ]	307
H.46 Inter-language Calling [DJS]	307
H.47 Dynamically-linked Code and Self-modifying Code [NYY]	308
H.48 Library Signature [NSQ]	308
H.49 Unanticipated Exceptions from Library Routines [HJW]	308
H.50 Pre-processor Directives [NMP]	309
H.51 Suppression of Run-time Checking [MXB]	309
H.52 Provision of Inherently Unsafe Operations [SKL]	309
H.53 Obscure Language Features [BRS]	309
H.54 Unspecified Behaviour [BQF]	310
H.55 Undefined Behaviour [EWF]	311
H.56 Implementation-defined Behaviour [FAB]	312

H.57 Deprecated Language Features [MEM] 312

Bibliography 313

Index 316

This document is a preview generated by EVS

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 [6.43 Inheritance \[RIP\]](#))
- Numerical analysis (although some simple items regarding the use of floating point are described in [6.5 Floating-point Arithmetic \[PLF\]](#))
- Inter-language operability

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 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*

3. Terms and definitions, symbols and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382-1 and the following apply. Other terms are defined where they appear in *italic* type.

3.1.1 Communication

3.1.1.1

protocol

set of rules and supporting structures for the interaction of threads

Note 1: A protocol can be tightly embedded and rely upon data in memory and hardware to control interaction of threads or can be applied to more loosely coupled arrangements, such as message communication spanning networks and computer systems.