ISO/IEC TS 17961:2013-11 (E)

Information technology - Programming languages, their environments and system software interfaces - C secure coding rules

Contents

Page

Foreword v

Introduction vi

1 Scope 1

2 Conformance 1
2.1 Portability assumptions 2

3 Normative references 2

4 Termsanddefinitions 2

5 Rules 5
5.1 Accessing an object through a pointer to an incompatible type [ptrcomp] 5
5.2 Accessing freed memory [accfree] 6
5.3 Accessing shared objects in signal handlers [accsig] 7
5.4 No assignment in conditional expressions [boolasgn] 8
5.5 Calling functions in the C Standard Library other than abort, _Exit, and signal from within a signal handler [asyncsig] 9
5.6 Calling functions with incorrect arguments [argcomp] 11
5.7 Calling signal from interruptible signal handlers [sigcall] 12
5.8 Calling system [syscall] 13
5.9 Comparison of padding data [padcomp] 14
5.10 Converting a pointer to integer or integer to pointer [intptrconv] 14
5.11 Converting pointer values to more strictly aligned pointer types [alignconv] 15
5.12 Copying a FILE object [filecpy] 16
5.13 Declaring the same function or object in incompatible ways [funcdecl] 16
5.14 Dereferencing an out-of-domain pointer [nullref] 18
5.15 Escaping of the address of an automatic object [addrescape] 18
5.16 Conversion of signed characters to wider integer types before a check for EOF [signconv] 19
5.17 Use of an implied default in a switch statement [swtchdflt] 19
5.18 Failing to close files or free dynamic memory when they are no longer needed [fileclose] 20
5.19 Failing to detect and handle standard library errors [liberr] 20
5.20 Forming invalid pointers by library function [libptr] 26
5.21 Allocating insufficient memory [insufmem] 28
5.22 Forming or using out-of-bounds pointers or array subscripts [invptr] 29
5.23 Freeing memory multiple times [dblfree] 34
5.24 Including tainted or out-of-domain input in a format string [usrfmt] 35
5.25 Incorrectly setting and using errno [inverrno] 37
5.26 Integer division errors [diverr] 39
5.27 Interleaving stream inputs and outputs without a flush or positioning call [ioileave] 40
5.28 Modifying string literals [strmod] 41
5.29 Modifying the string returned by getenv, localeconv, setlocale, and strerror [libmod] 42
5.30 Overflowing signed integers [intoflow] 43
5.31 Passing a non-null-terminated character sequence to a library function that expects a string [nonnullcs] 44
5.32 Passing arguments to character-handling functions that are not representable as unsigned char [chrsgnext] 45

5.33 Passing pointers into the same object as arguments to different restrict-qualified parameters [restrict] 46
5.34 Reallocating or freeing memory that was not dynamically allocated [xfree] 47
5.35 Referencing uninitialized memory [uninitref] 48
5.36 Subtracting or comparing two pointers that do not refer to the same array [ptrobj] 49
5.37 Tainted strings are passed to a string copying function [taintstrcpy] 50
5.38 Taking the size of a pointer to determine the size of the pointed-to type [sizeofptr] 50
5.39 Using a tainted value as an argument to an unprototyped function pointer [taintnoproto] 51
5.40 Using a tainted value to write to an object using a formatted input or output function [taintformatio] 52
5.41 Using a value for fsetpos other than a value returned from fgetpos [xfilepos] 52
5.42 Using an object overwritten by getenv, localeconv, setlocale, and strerror [libuse] 53
5.43 Using character values that are indistinguishable from EOF [chreof] 54
5.44 Using identifiers that are reserved for the implementation [resident] 55
5.45 Using invalid format strings [invfmtstr] 57
5.46 Tainted, potentially mutilated, or out-of-domain integer values are used in a restricted sink [taintsink] 58

Annex A (informative) Intra- to Interprocedural Transformations 59

Annex B (informative)UndefinedBehavior 63

Annex C (informative) Related Guidelines and References 71

Annex D (informative) Decidability of Rules 77

Bibliography 78