ISO/IEC JTC 1/SC 22/WG 14 N1662 - Commenting Template

To submit your comments, submit this spreadsheet using a filename with the following format: FML-yymmdd.xls where "FML" is your

Commentor's Initials	Comment #	Category (see the category tab)	Rule Code	Subsection	Page Number	Line Number	Comment and rationale	Proposed new text	Record of Response
DEW	1	TL		Introduction	vii		Why is this sentence in the introduction and not in section 2 about conformance? If it is to impact conformance, it needs to be moved into section 2. "Specification assumes that an analyzer's visibility extends beyond the boundaries of the current function or translation unit being analyzed (see Annex A (informative) Intra- to Interprocedural Transformations)."		The sentence "This Technical Specification assumes..." does not contain a conformity requirement; rather the sentence is an observation that can be deducted from the various rules in the TS and provides in the Introduction a link to Annex A. Therefore the right place for the sentence is in the Introduction.
DEW	2	TH		Completeness and Soundness, last paragraph	vii		"Analyzers are trusted processes, meaning that developers rely on their output. Consequently, developers must ensure that this trust is not misplaced. To earn this trust, the analyzer supplier should, ideally, run appropriate validation tests. Although it is possible to use a validation suite to test an analyzer, no formal validation scheme exists at this time." What is the purpose of this paragraph? In particular the third sentence is implying if not explicitly stating some sort of requirement upon the developer of the analyzer. To me it reads as a veiled requirement that Analyzers conforming to the specification must pass some sort of validation.	Remove the paragraph from the specification.	Paragraph removed.

DEW	3	TL		Security focus last sentence	vii		"Implementers are encouraged to distinguish violations that involve tainted values from those that do not involve tainted values." The term tainted value has not been introduced yet.	Suggested rewording: Implementers are encouraged to distinguish violations that involved data from an external source of untrusted data that might have come from a malicious users or attacker from data that is not from an external source of untrusted data.	Reworded as: Implementers are encouraged to distinguish violations that operate on untrusted data from those that do not.
DEW	4	TL		Taint and tainted sources	viii		Page viii, Under Tainted sources include, is a list of functions that are Tainted sources. Shouldn't this be in the normative section of the specification, instead of the introduction?	Move the list to 4.14 Tainted Sources Tainted sources include -- parameters to the main function, -- the returned values from localeconv, fgetc, getc, getchar, fgetwc, getwc, and getwchar, and -- the strings produced by getenv, fscanf, vfscanf, vscanf, fgets, fread, fwscanf, vfwscanf, vwscanf, wscanf, and fgetws.	List moved to 4.14 Tainted Sources
DEW	5	TH		4.5	4		"NOTE 2 Mutilated values cannot be sanitized." Why cannot a Mutilated value be sanitized? What normative text supports the contention expressed in this note? What is the purpose of this note? Does it mean that an analyzer cannot determine if a Mutilated value has been sanitized?	Remove the note. Or maybe the note should say: NOTE 2 Analyzers may not be able to determine if a Mutilated value has been sanitized.	Note has been removed.

DEW	6	TL		4.9	5		Shouldn't the cross references be using the unique section identifiers instead of the section numbers: "see 5.8, 5.14, 5.23, 5.29, 5.38, and 5.45" as per the second sentence of paragrpaph 8 of the introduction on page vi which says: "The unique section identifiers are mainly for use in identifying the rules should the section numbers change because of the addition or elimination of a rule."	If not, then the the second sentence of paragrpaph 8 of the introduction on page vi should be removed.	Changed the second sentence of paragrpaph 8 of the introduction to read "The unique section identifiers are mainly for use by other documents in identifying the rules should the section numbers change because of the addition or elimination of a rule. "
DEW	7	TH	[sidcall]	5.7 Calling signal from interruptible signal handlers	14		Reword "Calling signal from within a signal handler whose execution can be interrupted by receipt of a signal on platforms where signal handlers are non-persistent shall be diagnosed." to use the terminology in the terms and definitions section	"On systems with non-persistent signal handlers, calling signal from within a signal handler whose execution can be interrupted by receipt of a signal." And in the Example(s) change "implementations where signal handlers are non-persistent" to "implementations with non-persistent signal handlers".	Changes made.

DEW	8	TH	[intptrconv]	5.10 Converting a pointer to integer or integer to pointer	16		Example 1, fix the wording ...	Change "because the pointer ptr is converted to an integer" to "because the results of converting the 64-bit ptr cannot be represented in the 32-bit integer type. Or change the entire sentence to: In this noncompliant example, a diagnostic is required because the results of converting the 64-bit pointer cannot be represented in a 32-bit integer.	Changed to: EXAMPLE 1 In this noncompliant example, a diagnostic is required on an implementation where pointers are 64 bits and unsigned integers are 32 bits because the result of converting the 64-bit ptr cannot be represented in the 32-bit integer type.
DEW	9	TH	[intptrconv]	5.10 Converting a pointer to integer or integer to pointer	16		EXAMPLE 2 In this noncompliant example, a diagnostic is required because the integer literal 0xdeadbeef is converted to a pointer. should be reworded as:	EXAMPLE 2 In this noncompliant example, a diagnostic is required because the conversion of the integer literal 0xdeadbeef to a pointer results in a pointer that does not point to an entity of the referenced type.	Change made.
DEW	10	E	[alignconv]	5.11	17		fix the typo in the Rationale "thatn"		Typo fixed
DEW	11	TL	[funcdecl]	5.13	19		Add some explanation as to why bash_groupname_completion_function and bash_groupname_completion_funct might be identical on an implementation. I.e. refer to the section of the C standard that defines the minimum number characters for external names that an implementation must support. We don't need to leave this as an exercise for the reader.		Add comments to example the example that "the identifier exceeds 31 characters" and that "identifier not unique within 31 characters"
DEW	12	E	[usrfmt]	5.23	39		Remove the three random boxes.		All boxes removed from code examples.

DEW	13	TH	[usrfmt]	5.24	39		Example 1 and 2 The term "external catalog" is not defined, but the term "tainted source" is, and I'm pretty sure that is what is meant.	Example 1 and Example 2 change "external catalog" to "tainted source".	Change made.
DEW	14	TH	[usrfmt]	5.25	39		Example 4, Stick with terminology defined in the specification. The spec doesn't define what a "user" is ...	Example 4 change "which is not controlled by the user" to "which does not contain tainted values".	Changed to: EXAMPLE 4 In this compliant example, a diagnostic is not required because the argument fmt is constrained to be one of the elements of the formats array, none of which are tainted values.
DEW	15	TL	[inverrno]	5.24.2	42		Don't know what it means to clear errno. Note "set errno to zero" wording is used properly in 5.24.1.	Change "clearing errno" to "set errno to zero"	Changed to "without setting errno to zero"
DEW	16	TL	[inverrno]	5.24.3	42		There exists a footnote "a", but no references in the section to footnote a.	Remove the footnote.	Footnote removed from table 6
DEW	17	E	[diverr]	5.25	44		Typo in example 3 and 4	In both Example 3 and Example 4, change "can result" to "cannot result".	Changes made.
DEW	18	TH	[intoflow]	5.29	48		The Rationale says Signed integer overflow is undefined behavior. The reference UB table is missing.	Add the UB table	Added UB 36 An exceptional condition occurs during the evaluation of an expression (6.5).
DEW	19	TL	[chrsgnext]	5.31	51		Don't understand how anything is unrepresentable as an unsigned char. The point (after it was explained to me) is that the type being passed to these functions my represent a value that is not representable as an unsigned char. A simple rewording of the leadin wording will help significantly in understanding this issue.	Change the leadin for the Example to read: In this noncompliant example, a diagnostic is required because the parameter to isspace, *t is defined as a const char * which after promotion to an int may not be representable as an unsigned char.	Change made.

DEW	20	TH	[strmod]	5.27 Modifying string literals	46		Add an Exception Rationale for this change: Refer to 5.33 where a similar exception is given for implementations that cause a runtime-constraint violation when reallocating or freeing memory that was not dynamically allocated.	This violation does not need to be diagnosed on implementations that cause a runtime-constraint violation when modifying a string literal.	A “runtime-constraint violation” is only defined in Annex K. Presumably, the commenter meant “perform a trap”. The exception for 5.3 intended for implementations on which the error does not cause a trap, which is not the case here.
DEW	21	E	[uninitref]	5.34	53		Remove extraneous box in Rationale section		All boxes removed from code examples.
DEW	22	TH	[taintstrcpy]	5.36	56		What is a tainted string? Where is it defined?	I'd suggest changing "tainted string" to strings from a tainted source, or define tainted strings in the terms and definitions section. But first see my next comment	Changed to: "Tainted values that are referenced by the source argument to the strcpy, strcat, wcscpy, or wcscat function and that exceed the size of the destination array shall be diagnosed."
DEW	23	TH	[taintstrcpy]	5.37	56		Wording of the rule seems awkward. As written the rule applies for tainted strings that exceed the size of the destination array. Shouldn't this just say tainted strings?	Change the rule to say: Strings, wide or narrow, from a tainted source that are passed as the source argument to the strcpy, strcat, wcscpy, or wcscat function shall be diagnosed.	See DEW 21

DEW	24	TH	[resident]	5.43	63		EXAMPLE 9 In this compliant example, a diagnostic is not required because the reserved identifiers malloc and free are not used to define functions.
							
							The example that follows has no uses at all of the identifiers malloc and free. Either a better example is needed or the leadin should be changed.
								I'd suggest the example be changed to read as follows:
								
								static void *my_malloc(size_t nbytes) {
								  void *ptr;
								  /* ... */
								  /* allocate storage from own pool and set ptr */
								  return ptr;
								}
								
								static void free(void *ptr) {
								  /* ... */
								  /* return storage to own pool */
								}
									The example has been replaced with a new example.
									char *my_malloc(size_t nbytes) {
									  if (nbytes > 0) {
									    return (char *)malloc(nbytes);
									  } else {
									    return NULL;
									  }
									}
									
									void my_free(char *p) {
									  if(p != NULL) {
									    free(p);
									    p = NULL;
									  }
									}

DEW	25	E	[taintsink]	5.45	64		Remove the extraneous boxes, two occurances.		

DEW	26	TH	[taintsink]	5.46	64		Restricted sinks for integers are -- any pointer arithmetic, including array indexing; -- a length or size of an object (for example, the size of a variable-length array); -- the bound of access to an array (for example, a loop counter); and -- function arguments of type size_t or rsize_t (for example, an argument to a memory allocation function). These two definitions of Restrict sync conflict and need to normalized. I think 5.45 needs to be reworded.	For the first bullet I suggest: -- integer operands of any pointer arighmetic, including array indexing; For the second bullet I suggest the following to cover VLAs. But other than VLAs I don't understand what the original wording is trying to cover when it says "length or size of an object" -- The assignment expression for the declaration of a variable-length array For the third bullet, I suggest the following, but I'm not sure it covers everything that is intended (it covers using the operand in array subscripting): -- the postfix expresson preceding square brackets [] or the expression in square brackets [] of a subscripted designation of an element of an array object. The forth bullet is fine as is -- function arguments of type size_t or rsize_t (for example, an argument to a memory allocation function).	Section now reads: "Restricted sinks for integers include -- integer operands of any pointer arithmetic, including array indexing; -- the assignment expression for the declaration of a variable length array; -- the postfix expresson preceding square brackets [] or the expression in square brackets [] of a subscripted designation of an element of an array object; and -- function arguments of type size_t or rsize_t (for example, an argument to a memory allocation function)."
DEW	27	E		Annex C	80-88		Annex C is filled with text surrounded by spurious boxes. Remove the boxes.		All boxes removed from code examples.
