n2139

C Safe Secure Coding Rules Study Group

Robert C. Seacord

Agenda

Requirements of the safety and security markets

MISRA C and CERT C

History of ISO/IEC TS 17961

Combined safety and security International Standard

The C programming Language

Spirit of C:

a. Trust the programmer.
b. Do not prevent the programmer from doing what needs to be done.
c. Keep the language small and simple.
d. Provide only one way to do an operation.
e. Make it fast, even if it is not guaranteed to be portable.

The C programming language serves a variety of markets including safety-critical systems and secure systems.

While advantageous for system level programming, facets (a) and (b) can be problematic for safety and security.

Consequently, the C11 revision added a new facet:

f. Make support for safety and security demonstrable.

Coding Standards

Coding standard exist to service the safety and security markets performing C language development.

The safety-critical systems market is primarily served by The Motor Industry Software Reliability Association (MISRA), a UK-based collaboration between

- manufactures
- component suppliers
- engineering consultancies

The security market is primarily addressed by The CERT C Coding Standard published by Addison-Wesley.

Automotive and Aerospace Industries

The automotive and aerospace industries are major consumers of coding standards for safety-critical systems.

At many organizations, safety-critical code is written in C.

Extensive tool support for this language including

- strong source code analyzers
- logic model extractors
- metrics tools
- debuggers
- test support tools
- choice of mature, stable compilers.

Safety Standards

The safety community constrains development to a subset of the C language that is considered less prone to error.

These language subsets are influenced by the IEC 61508 series of international standards for electrical, electronic, and programmable electronic safety-related systems.

ISO 26262 Road vehicles — Functional safety

ISO 26262 is an adaptation of IEC 61508 for automotive electric/electronic systems that has been widely adopted by the major automotive manufacturers.

Focuses on the electronic systems installed in series-production passenger cars.

Introduces four Automotive Safety Integrity Levels (ASIL A – D)

- ASIL D is the most stringent level.
- allows different methods to be applied depending upon the ASIL of the system at a functional level.

Part 6 specifically addresses software development, placing requirements on

- the initiation of software development
- software architectural design and software unit design
- implementation

Design Principles for Software Unit Design and Implementation

M3CM = MISRA C:2012

Methods		ASIL				The Tool
		A	B	C	D	
1a.	One entry and one exit point in subprograms and functions	++	++	++	++	Applicable to M3CM rule 15.5
1b.	No dynamic objects or variables, or else online test during their creation	+	++	++	++	Applicable to M3CM rule 21.3
1c.	Initialization of variables	++	++	++	++	Applicable to M3CM rule 9.1
1d.	No multiple use of variable names	+	++	++	++	Applicable to M3CM rules 5.3, 5.6-5.9
1e.	Avoid global variables or else justify their usage	+	+	++	++	Applicable to M3CM rules 8.7, 8.9
1f.	Limited use of pointers	o	+	+	++	Applicable to M3CM rules 8.13, 18.1 – 18.5
1g.	No implicit type conversions	+	++	++	++	Applicable to M3CM rules 10.1-10.8, 11.6-11.9
1h.	No hidden data flow or control flow	+	++	++	++	Applicable to M3CM rules 2.1, 5.3,13.2, 18.1-18.5, 20.1, 20.7, 20.10, directive 4.9
1i.	No unconditional jumps	++	++	++	++	Applicable to M3CM rule 15.1
1j.	No recursions	+	+	++	++	Applicable to M3CM rules 17.2

Aviation Standards

DO-178C Software Considerations in Airborne Systems and Equipment Certification’ standard published by the Radio Technical Commission for Aeronautics (RTCA) is used for commercial software-based aerospace systems.

Security Standards

The security community serves a broader market

Security is more often an attribute of applications and systems whose primary purpose is to deliver functionality and for which security is typically one of several system qualities that may be traded-of against other qualities such as performance and usability.

These applications frequently make use of the whole language, including dynamic memory, which makes subsetting the language too costly to consider.

Agenda

Requirements of the safety and security markets

MISRA C and CERT C

History of ISO/IEC TS 17961

Combined safety and security International Standard

MISRA

The MISRA C Guidelines define a subset of the C language that reduces the opportunities for mistakes.

The first edition of MISRA C, ‘Guidelines for the use of the C language in vehicle based software’ was published in 1998 to provide a restricted subset of C to meet the requirements of IEC 61508 Safety Integrity Level 2 and above.

Since that time, MISRA C has been adopted by a wide variety of industries and applications including the rail, aerospace, military, and medical sectors.

The second edition, known as MISRA C:2004 is titled ‘Guidelines for the use of the C language in critical systems’.

The first two editions of MISRA were based on C90.

MISRA C:2012 extends support for C99 while maintaining guidelines for C90.

MISRA Provenance

MISRA started in the early 1990s as a project in the UK government's SafeIT programme.

The MISRA project was conceived to develop guidelines for the creation of embedded software in road vehicle electronic systems.

In November 1994, development guidelines for vehicle-based software were published.

Once the official funding had finished, the MISRA members continued to work together on an informal basis.

Today, the MISRA Consortium is coordinated by a steering committee of the member companies. The project management has been provided by MIRA Limited, a for-profit organization.

MISRA C Releases

MISRA C:2012

MISRA C:2012 Addendum 1
MISRA-C:2004 v MISRA C:2012

MISRA C:2012 Addendum 2
Coverage against “C Secure”

MISRA C:2012 Amendment 1
Additional security guidelines

MISRA C Work in Progress

Work In Progress – To be issued “soon”

- MISRA C:2012 Tech. Corr. 1
- MISRA C:2012 Addendum 2, 2nd Ed Coverage against “C-Secure”
- MISRA C:2012 Addendum 3 Coverage against “CERT-C”
- Exemplar Suite

To roll-up as a new issued MISRA C document in due course.

MISRA C

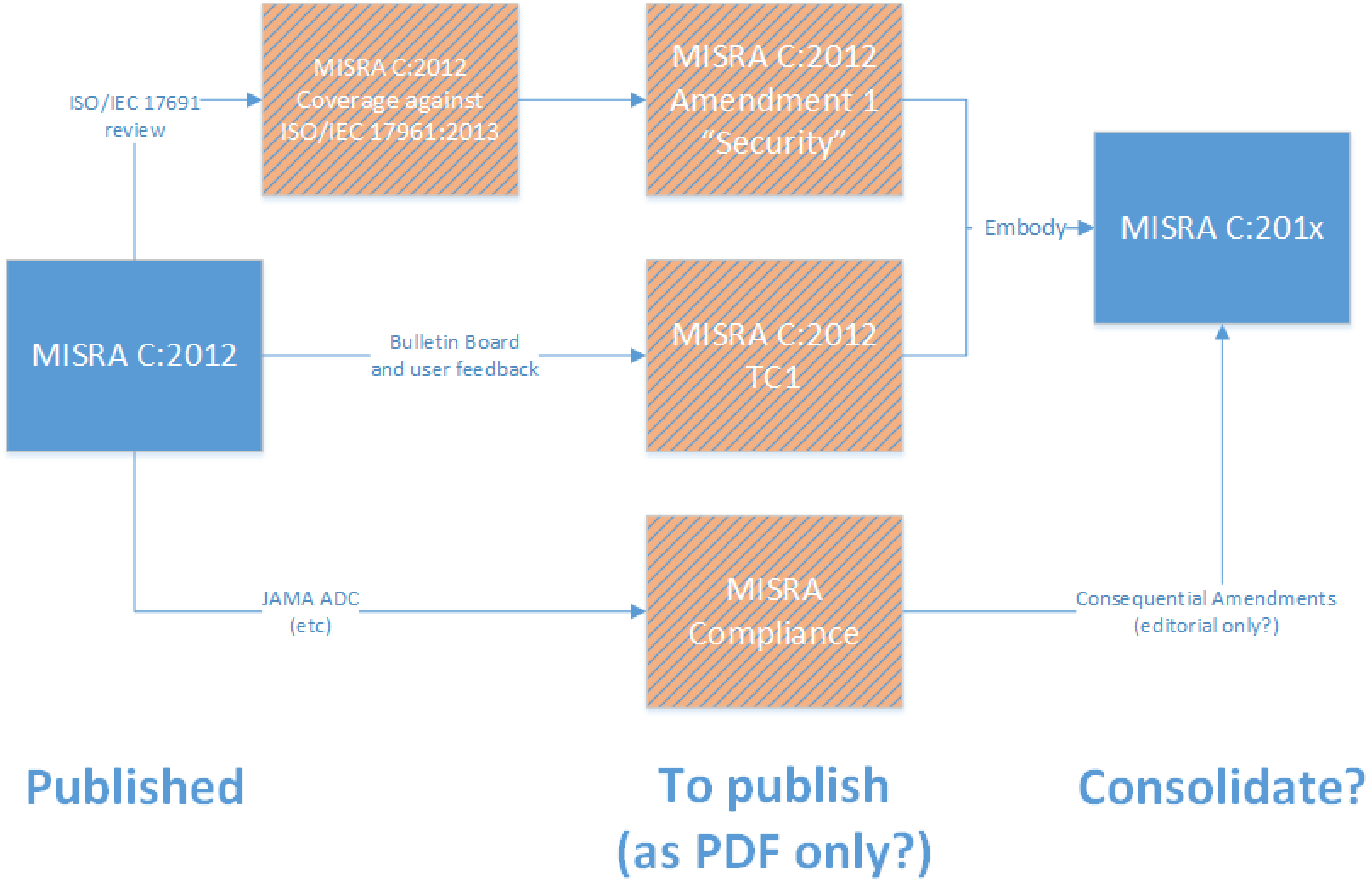
Work In Progress

- Update for C11 revisions to “Core” functionality
- Review of Standard Library rules and exclusions
- Enhancements to Exemplar Suite

New Work Items

- New C11 Functionality

MISRA Flowchart



CERT C Coding Standard

The CERT C Secure Coding Standard was developed at the request of, and in concert with, the C Standards Committee.

The 1st edition (a/ka/ CERT C:2008) was published in 14 October 2008.

The 2nd edition of The CERT C Coding Standard (a/k/a CERT C:2014) was

- published in 2014.
- updated to support C11
- aligned with ISO/IEC TS 17961

Hosted v Freestanding Implementations

The C Standard supports two forms of conforming implementations

- hosted
- freestanding
- In a freestanding environment, a C program execution may take place without the benefit of an operating system, as is common in low-end embedded systems.

MISRA C

- no library-specific restrictions on the subset of headers required in freestanding implementations
- major restrictions and prohibitions on many of the remaining standard headers in hosted implementations.

CERT C

- fully supports both hosted and freestanding environments

Key Features of Coding Standards

Coding Standard	C Standard	Security Standard	Safety Standard	International Standard	Whole Language
MISRA C:2004	C90	No	Yes	No	No
MISRA C:2012	C99	No	Yes	No	No
CERT C:2008	C99	Yes	No	No	Yes
CERT C:2014	C11	Yes	No	No	Yes
ISO/IEC TS 17961	C11	Yes	No	Yes	Yes

Standard Development Organizations

ISO/IEC JTC1/SC22/WG14 is the international standardization working group for the programming language C.

INCITS Technical Committee PL22.11 is the

- U.S. organization responsible for the C programming language standard.
- U.S. TAG to ISO/IEC JTC 1 SC22/WG14 and provides recommendations on U.S. positions to the JTC 1 TAG.

Agenda

Requirements of the safety and security markets

MISRA C and CERT C

History of ISO/IEC TS 17961

Combined safety and security International Standard

History

The idea of C secure coding guidelines arose during the discussion of the managed strings proposal at the Berlin meeting of the ISO/IEC JTC 1/SC 22/WG14 for standardization of the C language in March 2006.

The closest existing product at the time, MISRA C, was generally viewed by the committee as inadequate because, among other reasons, it precluded all the language features that had been introduced by ISO/IEC 9899:1999.

C Secure Coding Guidelines SG

WG14 established a study group to study the problem of producing analyzable secure coding guidelines for the C language.

- First meeting was held on October 27, 2009.
- Participants included analyzer vendors, security experts, language experts, and consumers.
- New work item approved March 2012; study group concluded.

ISO/IEC TS 17961

Applies to analyzers, including static analysis tools and C language compilers that wish to diagnose insecure code beyond the requirements of the language standard.

Enumerates secure coding rules and requires analysis engines to diagnose violations of these rules as a matter of conformance to this specification.

These rules may be extended in an implementation-dependent manner, which provides a minimum coverage guarantee to customers of any and all conforming static analysis implementations.

Secure Coding Validation Suite

A set of tests to validate the rules defined in TS 17961, these tests are based on the examples in this technical specification.

https://github.com/bluepilot/scvs

Distributed with a BSD-style license.

Agenda

Requirements of the safety and security markets

MISRA C and CERT C

History of ISO/IEC TS 17961

Combined safety and security International Standard

Safety/Security Rules Study Group

Following the Fall 2016 WG14 Pittsburgh a Safety/Security Rules Study Group was created to:

1. Study the problem of adding coverage for safety-critical and safety/security-critical issues into the existing C Secure Coding Rules TS.
2. Study the problem of addressing safety and security issues related to parts of the C standard not currently covered by the TS, such as concurrency.
3. Propose updates to TS 17961 based on these studies and based on experience gained with the TS since its publication.
4. Recommend to WG 14 a course of action for the resulting document, such as creating a new edition of the TS, or making it into an International Standard.

Study Group Members 1 of 3

• Roberto Bagnara	University of Parma / BUGSENG
• Aaron Ballman	GrammaTech
• Andrew Banks	Frazer Nash Research
• Jim MacArthur	Codethink
• Kayvan Memarian	University of Cambridge
• Clive Pygott	LDRA
• Robert C. Seacord	NCC Group / CMU
• Peter Sewell	University of Cambridge
• Barnaby Stewart	NCC Group
• Murali Somanchy	Qualcomm
• Daniel Godas-Lopez	Qualcomm

Study Group Members 2 of 3

• Elisa Heymann Pignolo	Universitat Autònoma de Barcelona
• Adele Carter	Kiteway
• Ian Hawkes	TRW
• Kayvan Memarian	University of Cambridge
• Chris Polin	Codethink
• Steve Christey Coley	MITRE
• Gavin McCall	Visteon Engineering Services
• Gerard Holzmann	Nimble Research
• Joe Jarzombek	Synopsys
• David Wheeler	IDA
• Konstantin Serebryany	Google

Study Group Members 3 of 3

• Barton Miller	University of Wisconsin-Madison
• Masaki Kubo	JPCERT
• Yozo Toda	JPCERT
• Michael Feiri	TRW
• Paul Sherwood	Codethink
• Bob Martin	MITRE
• Robin Randhawa	ARM
• Stephen Kell	University of Cambridge
• William Forbes	TRW

Working Document

Working document available on protected wiki:

https://gitlab.com/trustable/C_Safety_and_Security_Rules_Study_Group

Access available to members of the study group

Join the study group by sending email to rcseacord@gmail.com

Revision Schedule

The milestones and preliminary dates for the revision process are:

- Study group submits Draft 17961 IS to WG14 – March 2020
- CD Registration & Ballot (17961) — December 2021
- DIS Ballot (17961) — December 2022

This schedule allows for the formal adoption of a revised standard by the end of 2022, with a publication date of 2023.

References

[1] Seacord, R.: ‘The CERT C secure coding standard’ (Addison-Wesley, 2008)

[2] Seacord, R.: ‘The CERT C coding standard, second edition: 98 rules for developing safe, reliable, and secure systems’ (Addison-Wesley, 2014)

[3] Gerard, J.: ‘NASA/JPL laboratory for reliable software. 2006. The power of 10: rules for developing safety-critical code’, Computer, 2006, 39, (6), pp. 95–97, doi: http://www.dx.doi.org/10.1109/MC.2006.212

[4] IEC 61508:2010: ‘Functional safety of electrical/electronic/programmable electronic safety-related systems’, International Electrotechnical Commission, in 7 parts published in 2010

[5] ISO/DIS 26262 – Road vehicles – Functional safety. The standard consists of several parts, published in 2011

[6] DO-178C/ED-12C: ‘Software Considerations in Airborne Systems and Equipment Certification’, RTCA, 2011

[7] RTCA DO-332: ‘Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A’, December 2011

[8] Lockheed Martin: ‘Joint Strike Fighter Air Vehicle C++ Coding Standards for the system development and demonstration program’, Document Number 2RDU00001 Rev C.,December 2005. Available at http://www.stroustrup.com/ JSF-AV-rules.pdf, accessed 22 April 2016

References

[9] Checkoway, S., McCoy, D., Kantor, B., et al.: ‘Comprehensive experimental analyses of automotive attack surfaces’. D. Wagner (Chair), SEC’11, Proc. of the 20th USENIX Conf. on Security, San Francisco, CA, 8–12 August 2011. Available at http://www.usenix.org/events/sec11/tech/full_papers/Checkoway.pdf

[10] Hack the S. Available at http://www.su-tesla.space/2016/04/hack-s.html, accessed 19 April 2016

[11] Miller, C., Valasek, C.: ‘Remote exploitation of an unaltered passenger vehicle’, August 2015

[12] McCarthy, C., Harnett, K., Carter, A.: ‘Characterization of potential security threats in modern automobiles: a composite modelling approach’. Report no. DOT HS 812 074, National Highway Traffic Safety Administration, Washington, DC, October, 2014

[13] SAE J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems. Available at http://standards.sae.org/wip/j3061/

[14] Road Vehicles – Vehicle Cybersecurity Engineering. Available at standardsproposals.bsigroup.com/Home/Proposal/5410, accessed 21 April 2016

[15] MISRA (Motor Industry Software Reliability Association): ‘Guidelines for the use of the C language in vehicle based software’ (MIRA, Nuneaton, UK, 1998), ISBN 978-0-9524156-6-4