MISRA C:2023 Addendum 2

Coverage of MISRA C:2023 against ISO/IEC TS 17961 “C Secure”

October 2024

First published October 2024 by The MISRA Consortium Limited
1 St James Court
Whitefriars
Norwich
Norfolk
NR3 1RU
UK

www.misra.org.uk

ISBN 978-1-911700-14-2 PDF

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

MISRA C:2023 Addendum 2

Coverage of MISRA C:2023 against ISO/IEC TS 17961 “C Secure”

October 2024

MISRA Mission Statement

MISRA provides world-leading best practice guidelines for the safe and secure application of both embedded control systems and standalone software.

MISRA is a collaboration between manufacturers, component suppliers, engineering consultancies and academics which seeks to research and promote best practice in developing safety- and security-related electronic systems and other software-intensive applications.

To this end, MISRA conducts research projects and publishes documents that provide accessible information for engineers and management.

MISRA also facilitates the dissemination and exchange of information between practitioners through supporting and holding technical events.

Disclaimer

Adherence to the requirements of this document does not in itself ensure error-free robust software or guarantee portability and re-use.

Compliance with the requirements of this document, or any other standard, does not of itself confer immunity from legal obligations.

Foreword

While it is a widely held viewpoint that MISRA C provides best-practice guidelines for the development of safety-related systems, the publication by ISO/IEC JTC1/SC22/WG14 in 2013 of ISO/IEC 17961 *C Secure* [2] initiated discussion as to the applicability of MISRA C for secure applications.

In response, the MISRA C Working Group compiled this Addendum, which documents the coverage of MISRA C against *C Secure*.

This updated edition reflects the coverage of MISRA C:2023 [1] against ISO/IEC 17961, incorporating the 2016 technical corrigendum [3].

It is the view of the Working Group that MISRA C already provides the best best-practice guidelines for the development of critical systems, whether the focus be on safety or security, or a combination of both.

Andrew Banks FBCS CITP
Chairman, MISRA C Working Group

Acknowledgements

The MISRA C Working Group

The MISRA Consortium Limited would like to thank the following individuals for their significant contribution to the writing of this document:

Andrew Banks	LDRA Ltd (also Intuitive Consulting)
Jill Britton	Perforce
Clive Pygott	Columbus Computing Ltd

The MISRA Consortium Limited also wishes to acknowledge contributions from the following members of the MISRA C Working Group during the development and review process:

Dave Banham	BlackBerry Ltd
Gerlinde Kettl	Vitesco Technologies GmbH
Tibor Milić	Rimac Technology
Chris Miller	GE Aviation Ltd
Hamzath Pitchai Mohammed	Robert Bosch GmbH
Chris Tapp	Keylevel Consultants Ltd

The MISRA Consortium Limited also wishes to acknowledge contributions from the following individuals during the development and review process:

David Ward	HORIBA MIRA Limited

Other acknowledgements

DokuWiki was used extensively during the drafting of this document. Our thanks go to all those involved in its development.

This document was typeset using fonts licensed under the SIL Open Font License, Version 1.1:

- Open Sans — Copyright © 2020, The Open Sans Project Authors.

Contents

1 Introduction

1.1 Background

Throughout the development of MISRA C, the main focus has been to address vulnerabilities in the C language, particularly for use in embedded systems, and primarily targeted at safety-related applications. MISRA C particularly applies to freestanding applications, which use a subset of the C Standard Library.

One of the great successes of MISRA C has been its adoption across many industries, and in environments where safety-criticality is less of a concern, but where security is more of an issue.

The publication by ISO/IEC JTC1/SC22/WG14 in 2013 of ISO/IEC 17961:2013 *C Secure* [2] initiated discussion as to the applicability of MISRA C for secure applications. The MISRA C Working Group listened to those concerns, and have compiled this Addendum to document the coverage of MISRA C against *C Secure*.

This updated edition reflects the coverage of MISRA C:2023 [1] against ISO/IEC TS 17961 incorporating the 2016 technical corrigendum [3].

1.2 Applicability

This document provides a mapping of the guidance provided by ISO/IEC TS 17961:2013/Cor:2016 *C Secure* [3] against MISRA C.

This document should be read in conjunction with MISRA C:2023 *Guidelines for the use of the C language in critical systems* [1].

2 Coverage

2.1 Coverage classification

The coverage of each *C Secure* rule against MISRA C is classified as follows:

Status	Interpretation
Explicit	The *C Secure* rule is EXPLICITLY covered by one or more MISRA C guidelines, which directly addresses the undesired behaviour.
Implicit	The *C Secure* rule is IMPLICITLY covered by one or more MISRA C guidelines, although the behaviour is not explicitly referenced.
Restrictive	The *C Secure* rule is covered by one or more MISRA C guidelines that prohibit a language feature in a RESTRICTIVE manner.

2.2 Coverage strength

The strength of the coverage of each *C Secure* rule against MISRA C is classified as follows:

Status	Interpretation
Strong	The *C Secure* rule is covered by one or more targeted MISRA C Rules (excluding Rule 1.3 on its own).
Weak	The *C Secure* rule is only covered by one or more MISRA C Directives, or by Rule 1.3.
None	The *C Secure* rule is not covered by any MISRA C Guidelines.

3 C Secure cross reference

3.1 Guideline by Guideline

C Secure Rule	MISRA C:2023 Guidelines			Comments
	Guidelines	Coverage		
5.1	Rule 1.3, 10.8, 11.1, 11.2, 11.3, 11.5, 11.6, 11.7, 11.8	Explicit	Strong	
5.2	Dir 4.12 Rule 1.3, 21.3	Restrictive	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
5.3	Rule 1.3, 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
5.4	Rule 13.4	Explicit	Strong	MISRA C is stricter than C Secure.
5.5	Rule 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
5.6	Rule 1.3, 8.2, 17.3	Explicit	Strong	MISRA C requires all functions to be created with complete prototypes.
5.7	Rule 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
5.8	Rule 21.21	Explicit	Strong	MISRA C is stricter than C Secure as it prohibits the use of the function *system* of `<stdlib.h>`.
5.9	Rule 21.16	Explicit	Strong	
5.10	Rule 1.3, 11.4	Explicit	Strong	
5.11	Rule 11.3	Explicit	Strong	
5.12	Rule 22.5	Explicit	Strong	
5.13	Rule 1.3, 8.3, 8.4, 8.5	Explicit	Strong	
5.14	Dir 4.1 Rule 18.1	Explicit	Strong	
5.15	Rule 18.6	Explicit	Strong	
5.16	Dir 4.7 Rule 10.3, 22.7	Explicit	Strong	
5.17	Rule 16.4	Explicit	Strong	C Secure permits omission of *default* clause for enums if all conditions are covered.
5.18	Rule 22.1	Explicit	Strong	
5.19	Dir 4.7 Rule 17.7	Explicit	Strong	
5.20	Dir 4.1, 4.11 Rule 1.3	Implicit	Weak	MISRA C requires parameters passed to standard library functions are checked.
5.21	Dir 4.12 Rule 21.3	Restrictive	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
5.22	Rule 1.3, 18.1	Explicit	Strong	
5.23	Dir 4.12 Rule 1.3, 21.3	Explicit	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
5.24	Dir 4.1, 4.11, 4.14 Rule 1.3, 21.6	Implicit	Strong	The out-of-domain aspects of this rule are implicitly covered by Rule 1.3. In addition, MISRA C has a general prohibition on the use of the `<stdio.h>` I/O functions which catches issues with string formats.
5.25	Dir 4.1, 4.7, 4.11 Rule 22.8, 22.9, 22.10	Explicit	Strong	
5.26	Dir 4.1, 4.14 Rule 1.3	Implicit	Weak	
5.27	Dir 4.1 Rule 1.3, 21.6	Restrictive	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions.

C Secure Rule	MISRA C:2023 Guidelines			Comments
	Guidelines	Coverage		
5.28	Rule 7.4	Explicit	Strong	
5.29	Rule 1.3, 21.19	Explicit	Strong	
5.30	Dir 4.1, 4.14 Rule 1.3	Implicit	Weak	C Secure is only interested in overflow caused by taint.
5.31	Dir 4.1, 4.11 Rule 1.3	Implicit	Weak	
5.32	Dir 4.1, 4.11 Rule 1.3, 21.13	Explicit	Strong	
5.33	Rule 1.3, 8.14	Restrictive	Strong	MISRA C has a general prohibition on the use of the *restrict* keyword.
5.34	Rule 1.3, 22.2	Explicit	Strong	
5.35	Dir 4.12 Rule 1.3, 9.1, 9.7, 21.3, 22.14	Explicit	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation. Note: C Secure permits the use of uninitialized *unsigned char*.
5.36	Rule 1.3, 18.2, 18.3	Explicit	Strong	
5.37	Dir 4.1, 4.11 Rule 21.17	Explicit	Strong	
5.38	Rule 12.5	Explicit	Strong	
5.39	Rule 8.2	Explicit	Strong	MISRA C requires all functions to be created with complete prototypes.
5.40	Dir 4.1, 4.11, 4.14 Rule 1.3, 21.6	Restrictive	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions which catches issues with string formats.
5.41	Dir 4.1, 4.11 Rule 1.3, 21.6	Restrictive	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions
5.42	Rule 21.20	Explicit	Strong	
5.43	Rule 22.7	Explicit	Strong	
5.44	Rule 1.3, 20.4, 21.1, 21.2	Explicit	Strong	
5.45	Dir 4.1, 4.11 Rule 1.3, 21.6, 21.10	Restrictive	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions and `<time.h>` which catches issues with string formats.
5.46	Dir 4.1, 4.11, 4.14 Rule 1.3	Explicit	Weak	

3.2 Coverage summary

In summary, the coverage of MISRA C:2023 against *C Secure* is as follows:

Coverage	Strength	Number
Explicit	Strong	30
	Weak	1
Implicit	Strong	1
	Weak	4
Restrictive	Strong	10
	Weak	0
None	None	0
	Total	46 (of 46)

4 References

The following documents are referenced from within this amendment:

4.1 MISRA C

[1] MISRA C:2023 *Guidelines for the use of the C language in critical systems*
ISBN 978-1-911700-08-1 (paperback), ISBN 978-1-911700-09-8 (PDF),
The MISRA Consortium Limited, Norwich, April 2023

4.2 ISO/IEC JTC 1/SC 22/WG 14 Standards

[2] ISO/IEC TS 17961:2013
Information technology — Programming languages, their environments and system software interfaces — C secure coding rules

[3] ISO/IEC TS 17961:2013/Cor 1:2016
Information technology — Programming languages, their environments and system software interfaces — C secure coding rules

5 Change log

Date	ISBN	Revisions
April 2016	978-1-906400-15-6 PDF	Original release, for MISRA C:2012
January 2018	978-1-906400-18-7 PDF	Updated to reflect Amendment 1
October 2024	978-1-911700-14-2 PDF	Revised for MISRA C:2023
		Incorporate AMD3, AMD4