MISRA

MISRA C:2012 – Addendum 2

Coverage of MISRA C:2012 (including Amendment 1) against ISO/IEC TS 17961:2013 “C Secure”

2nd Edition, January 2018

First published January 2018 by HORIBA MIRA Limited
Watling Street
Nuneaton
Warwickshire
CV10 0TU
UK

www.misra.org.uk

ISBN 978-1-906400-18-7 PDF

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

MISRA C:2012 – Addendum 2

Coverage of MISRA C:2012 (including Amendment 1) against ISO/IEC TS 17961:2013 “C Secure”

2nd Edition, January 2018

MISRA Mission Statement

We provide 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 and engineering consultancies which seeks to promote best practice in developing safety- and security-related electronic systems and other software-intensive applications. To this end MISRA publishes documents that provide accessible information for engineers and management, and holds events to permit the exchange of experiences between practitioners.

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-critical systems, the publication by ISO/IEC JTC1/SC22/WG14 in 2013 of C Secure has initiated discussion as to the applicability of MISRA C for secure applications.

In response, the MISRA C Working Group have compiled this Addendum, which documents the coverage of MISRA C against C Secure. This second edition of the Addendum reflects the enhancements to MISRA C:2012 incorporated by the publication of Amendment 1.

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.

Andrew Banks FBCS CITP
Chairman, MISRA C Working Group

Acknowledgements

The MISRA consortium wishes to acknowledge the contribution made by the Japan Automobile Manufacturers Association to the preparation of this document.

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

Andrew Banks	Intuitive Consulting
Clive Pygott	Columbus Computing Ltd
Jill Britton	Programming Research Ltd

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

Fulvio Baccaglini	Programming Research Ltd
Dave Banham	Rolls-Royce plc
Paul Burden	Independent Consultant (Formerly Programming Research Ltd)
Mike Hennell	LDRA Ltd
Chris Hills	Phaedrus Systems Ltd
Chris Tapp	LDRA Ltd
Liz Whiting	LDRA Ltd

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

David Ward	HORIBA MIRA Ltd

Contents

1 Introduction 1

1.1 Glossary 1

1.2 Background 1

1.3 Changes from first edition 1

2 Coverage 2

2.1 Coverage classification 2

2.2 Coverage strength 2

3 ISO/IEC TS 17961 cross reference 3

3.1 Guideline by guideline 3

3.2 Coverage summary 6

4 References 7

1 Introduction

1.1 Glossary

In this document:

MISRA C	means the MISRA C:2012 Guidelines [1]
C Secure	means ISO/IEC 17961:2013 C Secure [2]
AMD1	means Amendment 1 to MISRA C:2012 Guidelines[3]

1.2 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 data-security is more of an issue.

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

1.3 Changes from first edition

The second edition adds coverage provided by Amendment 1 to MISRA C:2012 Guidelines.

The coverage summary has been updated to reflect the additional coverage.

2 Coverage

2.1 Coverage classification

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

Status	Interpretation
Explicit	The behaviour addressed by the C Secure rule is EXPLICITLY covered by one or more MISRA C guidelines, which directly addresses the undesired behaviour.
Implicit	The behaviour addressed by the C Secure rule is IMPLICITLY covered by one or more MISRA C guidelines, although the behaviour is not explicitly referenced.
Restrictive	The behaviour addressed by the C Secure rule is covered by one or more MISRA C guidelines that prohibit a language feature in a RESTRICTIVE manner. For example: • Rule 21.3 – `<stdlib.h>` (memory allocation/deallocation) • Rule 21.5 – `<signal.h>` (all) • Rule 21.6 – `<stdio.h>` (input/output functions) • Rule 21.8 – `<stdlib.h>` (getenv())
Partial/Restrictive	Some aspects of the behaviour addressed by the C Secure rule are covered in a RESTRICTIVE manner. However, some aspects of the behaviour are not covered by any MISRA C guidelines.
None	The behaviour addressed by the C Secure rule is not covered by any MISRA C guidelines.

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 behaviour addressed by the C Secure rule is covered by one or more targeted MISRA C rules.
Weak	The behaviour addressed by the C Secure rule is only covered by one or more MISRA C directives, or by Rule R1.3.
None	The behaviour addressed by the C Secure rule is not covered by any MISRA C guidelines.

Note: For C Secure rules with “partial” coverage, a combination of strength coverages is shown.

3 ISO/IEC TS 17961 cross reference

3.1 Guideline by guideline

C Secure Rule	MISRA C:2012			Comments
	Guidelines	Coverage		
Rule 5.01	Rule 1.3, 10.8, 11.2, 11.3	Explicit	Strong	
Rule 5.02	Dir 4.12 Rule 1.3, 21.3	Restrictive	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
Rule 5.03	Rule 1.3, 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
Rule 5.04	Rule 13.4	Explicit	Strong	*Note*: MISRA C is stricter than C Secure.
Rule 5.05	Rule 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
Rule 5.06	Rule 1.3, 8.2, 17.3	Explicit	Strong	MISRA C requires all functions to be created with complete prototypes.
Rule 5.07	Rule 21.5	Restrictive	Strong	MISRA C has a general prohibition on the use of `<signal.h>`.
Rule 5.08	Rule 21.8	Explicit	Strong	
Rule 5.09	Rule 21.6	Explicit	Strong	Coverage added with AMD1.
Rule 5.10	Rule 1.3, 11.4	Explicit	Strong	
Rule 5.11	Rule 11.3	Explicit	Strong	
Rule 5.12	Rule 22.5	Explicit	Strong	
Rule 5.13	Rule 1.3, 8.3, 8.4	Explicit	Strong	
Rule 5.14	Dir 4.1 Rule 18.1	Explicit	Strong	
Rule 5.15	Rule 18.6	Explicit	Strong	
Rule 5.16	Dir 4.7 Rule 10.3, 22.7	Explicit	Strong	Coverage added with AMD1.
Rule 5.17	Rule 16.4	Explicit	Strong	*Note*: C Secure permits omission of default clause for enums if all conditions are covered.
Rule 5.18	Rule 22.1	Explicit	Strong	

C Secure Rule	MISRA C:2012			Comments
	Guidelines	Coverage		
Rule 5.19	Dir 4.7 Rule 17.7	Explicit	Strong	
Rule 5.20	Dir 4.1, 4.11 Rule 1.3	Implicit	Weak	
Rule 5.21	Dir 4.12 Rule 21.3	Restrictive	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
Rule 5.22	Rule 1.3, 18.1	Explicit	Strong	
Rule 5.23	Dir 4.12 Rule 1.3, 21.3	Restrictive	Strong	MISRA C has a general prohibition on the use of dynamic memory allocation.
Rule 5.24	Dir 4.1, 4.11, 4.14 Rule 1.3, 21.6	Implicit	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions which catches issues with string formats. In addition, the out-of-domain aspects of this rule are implicitly covered by Rule 1.3, but MISRA C makes no explicit mention of taint. Coverage added with AMD1.
Rule 5.25	Dir 4.1, 4.7, 4.11 Rule 22.8, 22.9, 22.10	Explicit	Strong	Coverage added with AMD1.
Rule 5.26	Dir 4.1 Rule 1.3	Explicit	Weak	
Rule 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.
Rule 5.28	Rule 7.4	Explicit	Strong	
Rule 5.29	Rule 1.3, 21.8, 21.19	Explicit	Strong	Coverage added with AMD1.
Rule 5.30	Dir 4.1 Rule 1.3, 10.3, 10.4	Explicit	Weak	*Note*: C Secure is only interested in overflow caused by taint.
Rule 5.31	Dir 4.1, 4.11 Rule 1.3	Implicit	Weak	
Rule 5.32	Dir 4.1, 4.11 Rule 1.3, 21.13	Explicit	Strong	Coverage added with AMD1.
Rule 5.33	Rule 1.3, 8.14	Restrictive	Strong	MISRA C has a general prohibition on the use of the *restrict* keyword.

C Secure Rule	MISRA C:2012			Comments
	Guidelines	Coverage		
Rule 5.34	Rule 1.3, 22.2	Explicit	Strong	
Rule 5.35	Rule 1.3, 9.1	Explicit	Strong	*Note*: C Secure permits the use of uninitialised *unsigned char*.
Rule 5.36	Rule 1.3, 18.2, 18.3	Explicit	Strong	
Rule 5.37	Dir 4.1, 4.11 Rule 21.17	Explicit	Strong	Coverage added with AMD1.
Rule 5.38	Rule 12.5	Explicit	Strong	Coverage added with AMD1.
Rule 5.39	Rule 8.2	Explicit	Strong	MISRA C requires all functions to be created with complete prototypes.
Rule 5.40	Dir 4.1, 4.11, 4.14 Rule 21.6	Explicit	Strong	Coverage added with AMD1.
Rule 5.41	Dir 4.1, 4.11 Rule 1.3, 21.6	Implicit	Strong	
Rule 5.42	Rule 21.8, 21.20	Explicit	Strong	Coverage added with AMD1.
Rule 5.43	Rule 22.7	Explicit	Strong	Coverage added with AMD1.
Rule 5.44	Rule 1.3, 20.4, 21.1, 21.2	Explicit	Strong	
Rule 5.45	Dir 4.1, 4.11, 4.14 Rule 1.3, 21.6	Explicit	Strong	MISRA C has a general prohibition on the use of `<stdio.h>` I/O functions which catches issues with string formats. Coverage added with AMD1.
Rule 5.46	Dir 4.1, 4.11, 4.14 Rule 1.3	Explicit	Strong	Coverage added with AMD1.

3.2 Coverage summary

In summary, the coverage of MISRA C:2012 against C Secure is as follows:

Classification	Strength	Number
Explicit	Strong	32
	Weak	2
Implicit	Strong	2
	Weak	2
Restrictive	Strong	8
	Weak	0
Partial/Restrictive	Strong/None	0
None	None	0
	Total	46

4 References

[1] MISRA C:2012, *Guidelines for the use of the C language in critical systems*, ISBN 978-1-906400-10-1, MIRA Limited, Nuneaton, March 2013

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

[3] MISRA C:2012, *Amendment 1: Additional security guidelines for MISRA C:2012*, ISBN 978-1-906400-16-3, HORIBA MIRA Limited, Nuneaton, April 2016