The C rules and recommendations in this wiki are a work in progress and reflect the current thinking of the secure coding community. Because this is a development website, many pages are incomplete or contain errors. As rules and recommendations mature, they are published in report or book form as official releases. These releases are issued as dictated by the needs and interests of the secure software development community.

Create a sign-in account if you want to comment on existing content. If you wish to be more involved and directly edit content on the site, you still need an account, but you’ll also need to request edit privileges.

Front Matter

Page: Introduction

Rules

Page: Rule 01. Preprocessor (PRE)
Page: Rule 02. Declarations and Initialization (DCL)
Page: Rule 03. Expressions (EXP)
Page: Rule 04. Integers (INT)
Page: Rule 05. Floating Point (FLP)
Page: Rule 06. Arrays (ARR)
Page: Rule 07. Characters and Strings (STR)
Page: Rule 08. Memory Management (MEM)
Page: Rule 09. Input Output (FIO)
Page: Rule 10. Environment (ENV)
Page: Rule 11. Signals (SIG)
Page: Rule 12. Error Handling (ERR)
Page: Rule 13. Application Programming Interfaces (API)
Page: Rule 14. Concurrency (CON)
Page: Rule 48. Miscellaneous (MSC)
Page: Rule 50. POSIX (POS)
Page: Rule 51. Microsoft Windows (WIN)

Recommendations

Page: Rec. 01. Preprocessor (PRE)
Page: Rec. 02. Declarations and Initialization (DCL)
Page: Rec. 03. Expressions (EXP)
Page: Rec. 04. Integers (INT)
Page: Rec. 05. Floating Point (FLP)
Page: Rec. 06. Arrays (ARR)
Page: Rec. 07. Characters and Strings (STR)
Page: Rec. 08. Memory Management (MEM)
Page: Rec. 09. Input Output (FIO)
Page: Rec. 10. Environment (ENV)
Page: Rec. 11. Signals (SIG)
Page: Rec. 12. Error Handling (ERR)
Page: Rec. 13. Application Programming Interfaces (API)
Page: Rec. 48. Miscellaneous (MSC)
Page: Rec. 50. POSIX (POS)
Page: Rec. 51. Microsoft Windows (WIN)

CERT manifest files

As of 9/28/2018, the CERT manifest files are now available for use by static analysis tool developers to test their coverage of (some of the) CERT Secure Coding Rules for C, using many of 61,387 test cases in the Juliet test suite v1.2.

Back Matter

Page: AA. Bibliography
Page: BB. Definitions
Page: CC. Undefined Behavior
Page: DD. Unspecified Behavior
Page: EE. Analyzers
Page: FF. Related Guidelines
Page: GG. Risk Assessments
The CER C Coding Standard, 2016 Edition provides rules to help programmers ensure that their code complies with the new C11 standard and earlier standards, including C99. It is downloadable as a PDF. (errata)

Secure Coding in C and C++ identifies the root causes of today’s most widespread software vulnerabilities, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives.

SCALe offers conformance testing of C language software systems against the CERT C Secure Coding Standard.
**Contact Us**

Contact us if you
- have questions about the Secure Coding wiki
- have recommendations for standards in development
- want to request privileges to participate in standards development

**Thank You!**

We acknowledge the contributions of the following folks, and we look forward to seeing your name here as well.

---

**Rules vs. Recommendations**

This coding standard consists of rules and recommendations, collectively referred to as guidelines. Rules are meant to provide normative requirements for code, whereas recommendations are meant to provide guidance that, when followed, should improve the safety, reliability, and security of software systems. Learn more about the differences.

---

**Linking to Our Pages**

Link to guidelines using the Tiny Link under ToolsLink to this Page... (This URL will not change if the name of the guideline changes.)

---

**Information for Editors**

- To eliminate a section from the lists above, label it `section and void`.
- To have a section listed as a recommendation, label it `section and recommendation`.
- To have a section listed as a rule, label it `section and rule`.