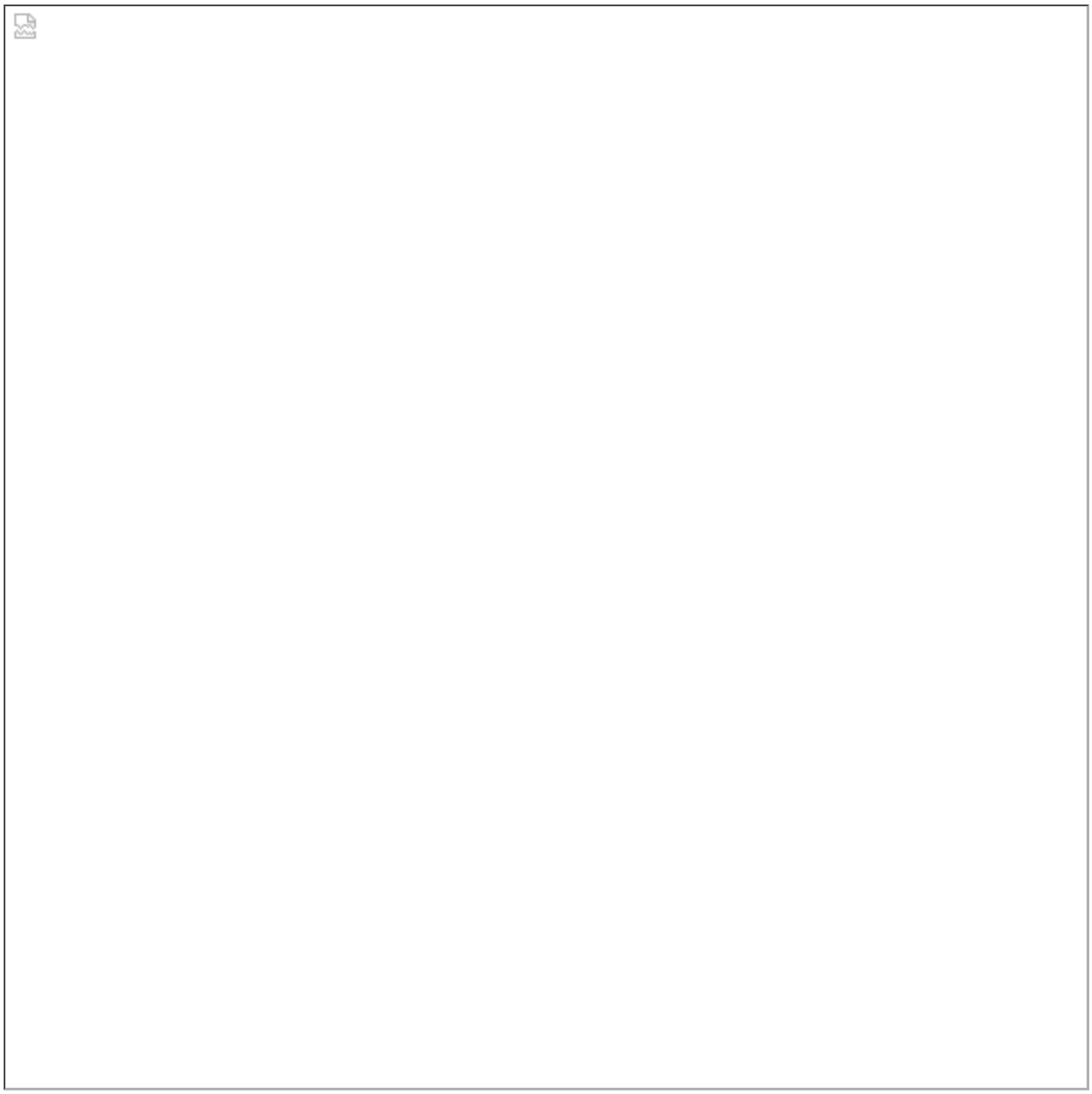

November 2013 Edition

News

We are still working hard to complete the CERT C Secure Coding Standard upgrade for C11. To do so, we need your help in reviewing the content and submitting comments on the wiki or by email. This is the last call for comments before publication.

Although we remain focused on security, we have begun to rename some of our publications to indicate that many of our coding standards go beyond security to address other quality attributes

as well. This broader scope is reflected in the title of our must recent book, Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs, and the upcoming revision to the CERT C Secure Coding Standard, which is tentatively titled The CERT C Coding Standard: 92 Rules for Developing Safe, Reliable, and Secure Systems, the tentative part being the number of rules. We hope you appreciate this direction as we expand our scope to address the broader range of issues our customers care about.

Our friend Milton Smith (Sr. Principal Security PM-Java Products, Java Platform Group Desk) brought the Java Platform Group, Product Management blog to our attention. All the product managers in the Java Platform Group, which includes Java SE, JavaFX, Java ME, and Javacard technology, blog here on topics primarily related to Java SE (generally), Java Security (broadly), Usability, and related areas of interest. If you don't already, you should also follow the CERT/Coordination Center blog, which may occasionally provide a counterpoint to Oracle's blog. CERT is committed to working with vendors such as Oracle to help resolve security issues, such as vulnerabilities, while remaining an independent voice.

If you would like to contribute to this or other efforts, and want to contact us privately, please send email to secure-coding@cert.org.

Top 10 Coding Guidelines for Java

Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs, is now available. This book was authored by a team of current and former CERT employees and visiting scientists, including Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, and David Svoboda. This same group of authors published *The CERT Oracle Secure Coding Standard for Java* in September 2011. The CERT website offers a support site for the book.

As a reader of this eNewsletter, we want to hear from you. How are you using CERT Secure Coding Standards? Write to us, and we may feature your work.

Language Standards Updates

CERT C Secure Coding Standard

Editors: Martin Sebor (Cisco Systems), Aaron Ballman (SEI)

Guidelines Added

INT19-C. Correctly compute integer widths for code that needs the number of width bits in an integer (which might be less than the integer's size). This rule is now cited by these rules:

CON07-C. Ensure that compound operations on shared variables are atomic
CON08-C. Do not assume that a group of calls to independently atomic methods is atomic
CON44-C. Wrap functions that can fail spuriously in a loop
DCL04-C. Do not declare more than one variable per declaration
INT32-C. Ensure that operations on signed integers do not result in overflow
INT34-C. Do not shift a negative number of bits or more bits than exist in the operand

Guidelines Changed

ARR37-C. Do not add or subtract an integer to a pointer to a non-array object
Added an exception for objects being an array of one element.

DCL01-C. Do not reuse variable names in subscopes
Renumbered the existing exception to start from 0 for consistency with other rules; added an exception allowing variable reuse in macros.

DCL22-C. Use volatile for data that cannot be cached
Changed from a rule to a recommendation; was previously listed as DCL34-C. Contents rewritten.

DCL39-C. Avoid information leak in structure padding
Updates applied during the course of review; serialization CS was rewritten.

ENV32-C. All exit handlers must return normally
Substantive updates applied during the course of review; now covers `quick_exit` handlers as well.

ERR30-C. Set errno to zero before calling a library function known to set errno, and check errno only after the function returns a value indicating failure
Updated risk assessment, corrected minor mistakes in table of functions.

EXP30-C. Do not depend on order of evaluation for side-effects
Updated and renamed during the course of review; was previously named "Do not depend on order of evaluation between sequence points."

ERR33-C. Detect and handle standard library errors
Merged the exceptions into a single exception; updated the list of exceptional functions.

EXP33-C. Do not read uninitialized memory
Removed UB from the Entropy compliant solution; reworded the rule for consistency during the course of review.

EXP40-C. Do not modify constant objects
Updated and renamed during the course of review; was previously named "Do not modify constant values."

FIO32-C. Do not perform operations on devices that are only appropriate for files
Updated the Windows CS to no longer use the `GetFileType()` API due to possible DoS concerns.

FLP06-C. Convert integers to floating point for floating-point operations
Changed from a rule to a recommendation; was previously listed as FLP33-C.

FLP32-C. Prevent or detect domain and range errors in math functions
Updated to include information about pole errors.

FLP36-C. Preserve precision when converting integral values to floating-point type
Renamed during the course of review; previously was named "Beware of precision loss when converting integral types to floating point."

INT30-C. Ensure that unsigned integer operations do not wrap
Updated the atomic integer CS wording to point out a potential race condition.

MEM30-C. Do not access freed memory
Updated to include wording about object lifetime; added an example of misusing automatic storage.

MEM31-C. Free dynamically allocated memory exactly once

MEM34-C. Only free memory allocated dynamically
Removed the exception allowing you to ignore deallocation of nonallocated memory when the library accepts such behavior.

STR30-C. Do not attempt to modify string literals
Updated during the course of review; the `strrchr()` CS was rewritten.

WIN03-C. Understand HANDLE inheritance
Updated with another NCCE/CS pair describing the behavior of the `fopen()` API on

Windows.

Guidelines Deprecated & Removed
ARR31-C. Use consistent array notation across all source files
Superseded by DCL40-C. Do not create incompatible declarations of the same function or object.
EXP38-C. Do not call offsetof() on bit-field members or invalid types
Only applicable to pre-standards K&R C; conforming C compilers issue errors in all circumstances.
MSC31-C. Ensure that return values are compared against the proper type
Superseded by INT31-C. Ensure that integer conversions do not result in lost or misinterpreted data, FLP37-C. Cast the return value of a function that returns a floating-point type, and INT18-C. Evaluate integer expressions in a larger size before comparing or assigning to that size.

CERT C++ Secure Coding Standard
Editors: Martin Sebor (Cisco Systems), Aaron Ballman (SEI)

No C++ rules were added, removed, deprecated, or substantively changed last month.

CERT Oracle Secure Coding Standard for Java
Editors: Adam O'Brien (Oracle), David Svoboda (SEI)

A New Java Section Was Added:

16. Java Native Interface (JNI)
It currently contains one rule:
JNI00-J. Define wrappers around native methods

Guidelines Changed
ENV00-J. Do not sign code that performs only unprivileged operations
Added a new exception to allow signed applets that specifically request to be sandboxed. This is supported in Java 7 since update 25.

No Java rules were removed last month

CERT Perl Secure Coding Standard
Editor: David Svoboda (SEI)

No Perl rules were added, removed, deprecated, or substantively changed last month.

Upcoming Events and Training

CERT STEPfwd (Simulation, Training, and Exercise Platform) combines extensive research and innovative technology to offer a new solution to cybersecurity workforce development-helping practitioners and their teams build knowledge, skills, and experience in a continuous cycle of

professional development. The goal of this approach is for cybersecurity professionals to use relevant knowledge, skills, and experience to successfully and effectively perform their duties-individually making improvements and collectively moving the organization forward.

FloCon
January 13-16, 2014
FloCon, a network security conference, takes place at the Francis Marion Hotel in Charleston, South Carolina. This open conference provides a forum for operational network analysts, tool developers, researchers, and other parties interested in the analysis of large volumes of traffic to showcase the next generation of flow-based analysis techniques. http://www.cert.org/flocon/

Mini-Track Announcement | Hawaii International Conference on Systems Sciences
Software Security for Mobile Platforms
January 6-9, 2014
Hilton Waikoloa, Big Island, Hawaii

What's The Best Way for a Programmer to Learn a New Language?
MOOCs? Books? Forums? Just diving in? Accomplished software developers, language creators, and authors share their perspectives on the best way to learn a new programming language. This article is part of our Learn a New Programming Language feature. (Includes a contribution from Robert Seacord).

Our People

Each month we highlight staff members behind our secure coding research. This month we feature Dan Plakosh and John Benito.

John Benito is an independent consultant providing software development, project management, and software testing. He is the current Convener of ISO/IEC JTC 1/SC 22/WG14, the ISO group responsible for Standard C, the initial Convener of ISO/IEC JTC 1/SC 22 WG 23 (was OWG Vulnerabilities), the project editor for the Technical Report 24772, and a member of the INCITS PL22.11 (ANSI C) technical committee. John previously was a member of INCITS PL22.16 (ANSI C++) and the ISO Java Study group. He has been in software development, project management, and testing for over 38 years. John has been participating in International Standard development for the past 24 years and is the recipient of the INCITS Exceptional International Leadership Award.

Dan Plakosh was the lead software engineer for the Systems Engineering

Department at the Naval Surface Warfare Center (NSWCDD) before joining the SEI. Dan has over 15 years of software development experience in defense, research, and industry. Dan's principal areas of expertise include real-time distributed systems, network communications and protocols, systems engineering, real-time 2D and 3D graphics, and UNIX OS internals. Much of Dan's recent experience has been redesigning legacy-distributed systems to use the latest distributed communication technologies.

Secure Coding Resources

Read *Supporting the Use of CERT Secure Coding Standards in DoD Acquisitions* by Timothy Morrow, Robert C. Seacord, John K. Bergey, and Philip Miller.

Listen to *Securing Mobile Devices aka BYOD* featuring Joe Mayes and Julia Allen.

Read *Secure Design Patterns* by Chad Dougherty, Kirk Sayre, Robert C. Seacord, David Svoboda, and Kazuya Togashi.

Subscribe to Our eNewsletter

Join the SEI CERT Secure Coding Community