Winter Edition 2015

News

The holiday season is upon us, even though Pittsburgh is still warm enough outside to open our windows and let fresh air circulate indoors.

The Clang community recently accepted a revision by Aaron Ballman to `clang-tidy`, allowing CERT checkers by name. The `clang-tidy` tool is used by developers when they wish to enable all diagnostic checks for their code (including compiler warnings, static analysis, and AST analysis) and is shipped as part of the Clang release. Starting with the Clang 3.8 release, users will be able to request diagnostics by their CERT guideline id, such as `-checks=cert-dcl54-cpp`, or request diagnostics for any CERT guideline via `-checks=cert-*`. For instance, instead of enabling `-checks=misc-new-delete-overloads` to test for compliance with DCL54-CPP. Overload allocation and deallocation functions as a pair in the same scope, the same check can be enabled through `-checks=cert-dcl54-cpp`.

As a result, the Secure Coding Initiative is pleased to announce new sections to the CERT C and C++ Coding Standards. These sections are called **New Clang Checkers** and they indicate rules that have new checkers in Clang that enforce them.

Lori Flynn co-chaired the Mobile Development Lifecycle workshop at SPLASH 2015, coordinating the panel "Smartphone Security" and chairing Session I of the presentations.

Lori Flynn and Will Klieber wrote the article, Smartphone Security, that was published in IEEE Pervasive Computing.

David Keaton and Dan Plakosh attended the C Standards Committee (ISO/IEC JTC 1/SC 22/WG 14) meeting on October 26-29 in Kona, HI, with David chairing the meeting and Dan

taking charge of the documents. The committee decided to publish a new edition of the C Secure Coding Rules, ISO/IEC TS 17961:2013. The new edition improves a few code examples that illustrate the rules. The ISO/IEC TS 17961:2013 is related to the CERT C Coding Standards, but the modified examples in ISO/IEC TS 17961:2013 are not used in the CERT C Coding Standards. So we don't intend any changes to the CERT C Coding Standards related to these changes of the ISO/IEC TS 17961:2013.

Aaron Ballman attended the C++ Standards Committee (ISO/IEC JTC 1/SC 22/WG 21) meeting on Oct 19-24 in Kona, HI, where he spent most of his time reviewing core C++ language wording for the anticipated release of C++17. For a broad overview of what's happening in C++, please see this write-up by committee member Stephan T Lavavej.

David Svoboda gave a 4-hour Secure Coding tutorial at the SEI's Software Solutions Conference.

How are you using the CERT Secure Coding Standards?

As a reader of this eNewsletter, your input is important to us. Submit your comments and let us know how you are using CERT Secure Coding Standards.

Language Standards Updates

CERT C Coding Standard

Editors: Martin Sebor (Red Hat, Inc.) and Aaron Ballman (SEI/CERT)

Changed

- DCL19-C. Minimize the scope of variables and functions
  Fixed a typo and added a cross reference
- EXP03-C. Do not assume the size of a structure is the sum of the sizes of its members
  Improved the wording of the Risk Assessment section
- FLP32-C. Prevent or detect domain and range errors in math functions
  Corrected a typo in the table (asin() was changed to asinh()) and corrected the domain expression for log1p() to correctly include the value -1 in the domain
- INT10-C. Do not assume a positive remainder when using the % operator
  Changed to a High severity (in case incorrect remainder produces out-of-bounds memory write)
- INT04-C. Enforce limits on integer values originating from tainted sources
  Improved the wording of the Risk Assessment section
- INT13-C. Use bitwise operators only on unsigned operands
  Modified the first exception to allow enumeration constants, as well as macros, to be involved in bitwise AND and OR operations
- ERR33-C. Detect and handle standard library errors
  Corrected the NCCE and CS for calloc() to focus on only a single type of failure, instead of having multiple failures present and only one addressed
- ARR38-C. Guarantee that library functions do not form invalid pointers
  Updated the NCCE and CS for two pointers + one integer to focus on the contents of the rule, instead of an unrelated recommendation

New Clang Checkers

- DCL03-C. Use a static assertion to test the value of a constant expression
- FIO38-C. Do not copy a FILE object

CERT C++ Secure Coding Standard

Editors: Martin Sebor (Red Hat, Inc.) and Aaron Ballman (SEI/CERT)

Added

- MEM57-CPP. Avoid using new expression for over-aligned types
- OOP58-CPP. Copy operations must mutate only the destination of the copy

Changed

- MEM54-CPP. Provide placement new with properly aligned pointers to sufficient storage capacity
  Added a non-compliant example demonstrating a failure to account for an array of new overhead along with a compliant solution

New Clang Checkers

- OOP11-CPP. Do not copy-initialize members or base classes from a move constructor
- DCL50-CPP. Do not define a C-style variadic function
- DCL54-CPP. Overload allocation and deallocation functions as a pair in the same scope
- ERR52-CPP. Do not use setjmp() or longjmp()
- ERR53-CPP. Do not reference base classes or class data members in a constructor or destructor function-try-block handler
- ERR58-CPP. Constructors of objects with static or thread storage duration must not throw exceptions
- ERR60-CPP. Exception objects must be nothrow copy constructible
- ERR61-CPP. Catch exceptions by lvalue reference

CERT Oracle Secure Coding Standard for Java

Editors: Brad Senetza (Oracle) and David Svoboda (SEI/CERT)

Added

- SER12-J. Prevent deserialization of untrusted classes
- SER13-J. Treat data to be deserialized as potentially malicious by default
  Added in response to CERT VU#576313 (aka CVE-2015-3253)

Changed

- STR00-J. Don't form strings containing partial characters from variable-width encodings
  Fixed code examples to properly build a string from the data read
- FIO10-J. Ensure the array is filled when using read() to fill an array
  Changed to emphasize the point that the array read methods of `java.io.InputStream` may read less than the expected number of bytes and to add a second noncompliant code example that uses the 3-argument read method
- FIO52-J. Do not store unencrypted sensitive information on the client side
  Modified the compliant solution to use `setHttpOnly()` and `setSource()` to protect the session ID

No Java rules were removed.

CERT Secure Coding Standard for Android
Editors: Fred Long (Aberystwyth University) and Lori Flynn (SEI/CERT)

No Android rules were added, removed, deprecated, or substantively changed.

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

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

Upcoming Events

Conference: Lori Flynn is co-chairing the research track of MobileSoft 2016, the 3rd ACM International Conference on Mobile Software Engineering and Systems (May 16-17) in Austin, TX (USA). All who are doing research on development, testing, and security related to mobile software are encouraged to submit a paper. The submission deadline is January 11, 2016. The MobileSoft conference is co-located with ICSE 2016 (May 14-22).

Our People
In the eNewsletter, we highlight staff members behind our secure coding research. This month we feature Will Snavely.

Hello, my name is Will Snavely. I recently graduated from Carnegie Mellon University (CMU) with my Master's in Information Security through the Information Networking Institute. Before joining the Secure Coding Team full time this November, I was an intern with the team for three semesters. I grew up in Tucson, Arizona, and attended the University of Arizona for my undergraduate education in computer science and mathematics. Before coming to CMU, I was an engineer at Microsoft in Redmond, Washington. My hobbies include juggling and playing music.

Secure Coding Resources

Read "Smartphone Security," in the IEEE Pervasive Computing Oct.-Dec. 2015 edition by Lori Flynn and Will Klieber

Read Adding Red to Blue: 10 Tactics Defenders Can Learn from Penetration Testers

Watch Using DidFail to Analyze Flow of Sensitive Information in Sets of Android Apps

Subscribe to Our eNewsletter

Join the SEI CERT Secure Coding Community