June / July 2014

News

A lot has happened so far this summer. You may have seen the recent Secure Coding Update concerning CERT's new tool. DidFail analyzes sets of Android apps for the leakage of sensitive information from a sensitive source to a restricted sink. The tool is free of charge and available to the public for download.

Will Klieber presented Android Taint Flow Analysis for App Sets (slides) at the ACM SIGPLAN

International Workshop on the State of the Art in Java Program Analysis in June.

Amar Bhosale graduated, with a nice Master's thesis, Precise Static Analysis of Taint Flow for Android Application Sets, describing our Android taint flow analysis, in depth.

David Svoboda and Robert Seacord's presentation Inside the CERT Oracle Secure Coding Standard for Java [CON2368] was accepted at JavaOne 2014. Along with Yozo TODA, Lead Analyst at the JPCERT Coordination Center, David also had a second talk accepted: Anatomy of Another Java Zero-Day Exploit [CON2120]. We are looking forward to another successful JavaOne conference this year.

The SEI report titled *Improving the Automated Detection and Analysis of Secure Coding Violations* has been published on the SEI website. This paper describes the approach used to add the ability to collect and statistically analyze data regarding coding violations and tool characteristics along with the initial results. The collected data will be used over time to improve the effectiveness of the SCALe analysis.

Aaron Ballman has returned from a successful C++ Standards meeting in Rapperswil, Switzerland, with big plans to update the CERT C++ Secure Coding Standard, which are being formulated on the C++ Coding Standard Development Guidelines page. Please feel free to join the discussion as we plan this major update.

Carol Lallier has finished retrofitting the off-line changes to *The CERT C Coding Standard,* Second Edition with the wiki, so the wiki now contains the fully synchronized "in development" version of this coding standard. The book remains the official standard against which SCALe assessments will be performed.

We continue to develop the Android secure coding standard on the Android Secure Coding wiki, and thanks go to everyone who has contributed with helpful comments! If you are an Android, Java, or C expert and would like to also contribute to completing this standard, we would be grateful for your input.

Otherwise, we have been quite busy with Source Code Analysis Laboratory (SCALe) assessments, which has led to a smattering of improvements to The CERT Oracle Secure Coding Standard for Java as we evolve rules to be clearer and more precise and to simplify conformance. Many of these changes are listed in the Java section below.
Please enjoy the rest of your summer-get out there and work on your tans!

How are you using the CERT Secure Coding Standards?

As a reader of this eNewsletter, we want to hear from you. Submit your comments about how you are using CERT Secure Coding Standards.

Language Standards Updates

CERT C Secure Coding Standard

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

Changed

- INT04-C. Enforce limits on integer values originating from tainted sources and ARR38-C. Guarantee that library functions do not form invalid pointers now both contain descriptions of the OpenSSL "Heartbleed" vulnerability, which violates both guidelines.
- The introductory code example in FLP32-C. Prevent or detect domain and range errors in math functions now uses `/* Handle error */` to indicate an error situation, which is our convention.
- The compliant solution in INT15-C. Use intmax_t or uintmax_t for formatted IO on programmer-defined integer types now uses `strtoumax()`, in accordance with INT06-C. Use strtol() or a related function to convert a string token to an integer.
- We added a solution to PRE11-C. Do not conclude macro definitions with a semicolon that uses an inline function, as recommended by PRE00-C. Prefer inline or static functions to function-like macros.

CERT C++ Secure Coding Standard

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

Changed

DCL33-CPP. Never qualify a variable of reference type with const or volatile
Corrected examples and updated text about Visual Studio diagnostics were added.

CERT Oracle Secure Coding Standard for Java

Editors: Adam O'Brien (Oracle), David Svoboda (SEI)

Added

- MSC08-J. Do not store non-serializable objects as attributes in an HTTP session
- IDS14-J. Do not trust the contents of hidden form fields
- IDS15-J. Do not allow sensitive information to leak outside a trust boundary
- VNA06-J. Do not use non-static member fields in a servlet
- These rules are all currently stubs. We will flesh them out soon.

Changed

- NUM12-J. Ensure conversions of numeric types to narrower types do not result in lost or misinterpreted data We removed the exception but included the use of explicit narrowing conversions using the remainder % operator in a compliant solution.
- OBJ03-J. Prevent heap pollution now permits mixing of generic types and raw types (it used to forbid these), as long as heap pollution does not occur.
- IDS09-J. Specify an appropriate locale when inspecting locale-dependent data We overhauled the introduction and added several code example pairs clarifying precisely what is compliant and what is not.

Deprecation Candidates

- NUM04-J. Do not use floating-point numbers if precise computation is required is conditional on programmer's intent
- NUM05-J. Do not use denormalized numbers is unenforcable.

- NUM06-J. Use the strictfp modifier for floating-point calculation consistency across platforms is conditional on programmer's intent.

CERT Secure Coding Standard for Android

- Analysis of Android Applicability: CERT's Java Coding Guidelines: Analyses with respect to guidelines 16, 17, and 19 were modified with text augmenting the guideline as written in the book so that the applicability to Android of the guideline is explained.
- DRD13-J. Do not provide addJavascriptInterface method access in a WebView which could contain untrusted content. (API level JELLY_BEAN or below) was scoped and titled to clarify that it is about Web Views in mobile apps.
- Analysis of Android Applicability: The CERT C Coding Standard (Rules and Recommendations) is a table listing the CERT C Coding Standard rules and recommendations, and states their applicability to the development of Android applications, according to an initial analysis.

CERT Perl Secure Coding Standard

Editor: David Svoboda (SEI)

No Perl rules were added, removed, deprecated, or substantively changed in May and June.

Upcoming Events and Training

Course:
Secure Coding in C and C++, August 19-22, 2014 (SEI, Pittsburgh, PA)

Secure Coding in C and C++ provides practical advice on secure practices in C and C++ programming. Producing secure programs requires secure designs. However, even the best designs can lead to insecure programs if developers are unaware of the many security pitfalls inherent in C and C++ programming. This course provides a detailed explanation of common programming errors in C and C++ and describes how these errors can lead to code that is vulnerable to exploitation.

Conference:
48th Annual Hawaii International Conference on System Sciences

January 5-8, 2015 Grand Hyatt Kauai, Hawaii

Conference:
TSP Symposium - See the Lineup of Speakers

The Team Software Process (TSP) Symposium 2014 technical program will go beyond the core methodology of TSP to encompass a broader range of complementary practices that contribute to peak performance on system and software projects.

The unifying theme of the conference is quality. Ultimately, a quality product and service must be delivered on time and within budget, be secure, be sustainable, and provide value to end users.

Our People

In the eNewsletter, we highlight staff members behind our secure coding research. This month we feature Robert Seacord.

Robert C. Seacord is the Secure Coding Technical Manager. He is the author of The CERT C Secure Coding Standard (Addison-Wesley, 2014) and Secure Coding in C and C++ (Addison-Wesley, 2002) as well as co-author of two other books. Robert is also an adjunct professor at Carnegie Mellon University and a technical expert for ISO/IEC JTC1/SC22/WG14, the international standardization working group for the programming language C.

Secure Coding Resources

Listen to *Characterizing and Prioritizing Malicious Code* by Jose A. Morales

Watch *SEI Panel Discussion on Heartbleed*

Read *Secure Coding to Prevent Vulnerabilities* by Robert Seacord

Subscribe to Our eNewsletter

Join the SEI CERT Secure Coding Community