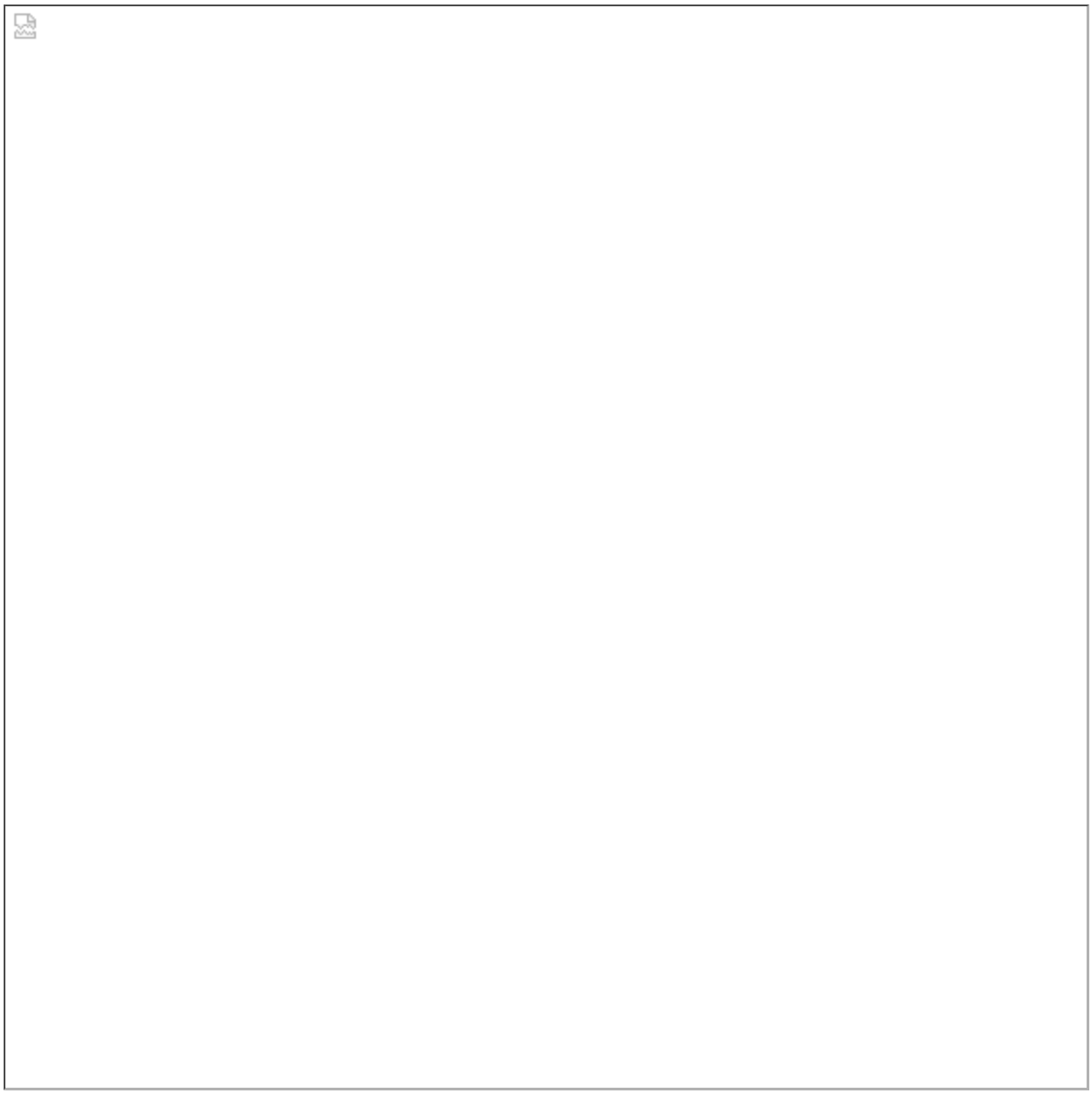

October 2013 Edition

News

The school year is well underway, so David Svoboda, Dean Sutherland, and Robert Seacord all escaped to the JavaOne 2013 conference where we enjoyed San Francisco and gave several talks:

- Session ID: CON6396 Don't Be Pwned: A Very Short Course on Secure Programming in Java, Dean Sutherland and Robert Seacord

- Session ID: CON3122 Anatomy of a Java Zero-Day Exploit, David Svoboda
- Session ID: TUT5599The Java Security Architecture: How and Why, David Svoboda

All three presentations were well attended, well-received, and recorded. These recordings will be made available on the JavaOne website soon.

JavaOne contained many fascinating presentations about the use of Java in the cloud, on the desktop, and in embedded devices. One of the new features of Java 8, lambda functions, has many people excited and will doubtless provide some interesting security ramifications. There were also many talks about alternative languages, such as Scala, running on the JVM.

Our presentations were part of a new track: *Securing Java*. This track contained many other fascinating talks ranging from malware analysis to upcoming security features in Java 8.

Security in Java is a top priority at Oracle, who has delayed the release of Java 8 to address a number of security issues in Java 7. Oracle has invited David Svoboda from CERT to meet with Oracle's security staff at an event organized for its Security Customer Advisory Council (SCAC), an advisory board established by Oracle to discuss security assurance policies and practices, including Java platform security concerns and mitigation practices.

CERT is also promoting *Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs*, which was the best-selling book at the JavaOne conference. Our previous book, *The CERT Oracle Secure Coding Standard for Java*, sold out at the conference book store.

We are still working hard to complete the second edition of the CERT C Secure Coding Standard, which will be published in a forthcoming published by Addison-Wesley and available in Spring, 2014. To do so, we need your help in reviewing the content and submitting comments on the wiki or by email. Please provide your comments by as as soon as possible, so that we will have time to incorporate them before publication. 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.

New Feature! 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

PRE13-C Use the Standard predefined macros to test for versions and features
Discusses the proper usage of predefined macros

Guidelines Changed

MSC23-C. Beware of vendor-specific library and language differences
Added language-specific entries for Microsoft Visual Studio 2012
EXP39-C. Do not access a variable through a pointer of an incompatible type
Added an example using realloc() to change types
EXP41-C. Do not add or subtract a scaled integer to a pointer
Added compliant solutions; removed "under construction" designation
CON00-C. Avoid race conditions with multiple threads
Fixed compliant solution using C11 atomics; thanks to Rajan Bhakta!
WIN04-C. Consider encrypting function pointers
Moved from the MSC section, updated compliant solution and wording
EXP33-C. Do not reference uninitialized memory
Added an exception for referencing uninitialized memory through an unsigned char
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
Added C11 content, rearranged much of the existing content
MSC37-C. Ensure that control never reaches the end of a non-void function
Added an exception for control reaching the end of main().
CON04-C. Join or detach threads even if their exit status is unimportant
Was previously listed as CON40-C as a rule; moved to recommendation due to enforceability.
CON05-C. Do not perform operations that can block while holding a lock
Was previously listed as CON36-C as a rule; moved to recommendation due to practicality.
FIO20-C. Avoid unintentional truncation when using fgets() or fgetws()
Was previously listed as FIO36-C as a rule; moved to recommendation due to enforceability.
Rewrote the NCCE and CS to be more directly related to the recommendation.
ARR38-C. Guarantee that library functions do not form invalid pointers
Substantially modified for clarity and accuracy.

Guidelines Deprecated & Removed

MEM08-C. Use realloc() only to resize dynamically allocated arrays
Deprecated due to coverage by EXP39-C. Do not access a variable through a pointer of an incompatible type
EXP04-C. Do not perform byte-by-byte comparisons involving a structure
Deprecated due to coverage by EXP42-C. Do not compare padding data
MEM32-C. Detect and handle memory allocation errors
Removed due to coverage by ERR33-C. Detect and handle standard library errors
FIO04-C. Detect and handle input and output errors
Removed due to coverage by ERR33-C. Detect and handle standard library errors
ERR31-C. Don't redefine errno
Removed due to coverage by DCL37-C. Do not declare or define a reserved identifier and MSC38-C. Do not treat as an object any predefined identifier that might be implemented as a macro
SIG33-C. Do not recursively invoke the raise() function
SIG32-C. Do not call longjmp() from inside a signal handler
Removed due to coverage by SIG30-C. Call only asynchronous-safe functions within signal

handlers

Resolved Issues
After a short teleconference, we determined that EXP35-C is not obsolete with respect to changes to C11. C11 introduced the term *temporary lifetime*, which actually increases the lifetime of a temporary object. We will be updating EXP35-C to use the new language and also to explain any differences in behavior between C11 and C99.

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)

Guidelines Added
DRD03-J. Do not broadcast sensitive information using an implicit intent
Three other Android rules are under development

No Java rules were removed last month.

Guidelines Changed
There are several improvements to
IDS04-J. Safely extract files from ZipInputStream:

The NCCE now uses a `finally` clause to ensure that the ZIP file is closed even if an exception is thrown. (The CS already did this.)

The CS now also counts the number of files extracted and throws an exception if the number exceeds a constant-1024 in the example.

The CS no longer extracts more than `total` bytes. This should address issues of running out of disk space, assuming it has space for `total` bytes. Actually the maximum amount of space required will be something like `total + BUFFER + TOOMANY * BLOCK_SIZE`. Exhausting disk space is a nasty problem, partially because many systems give no indication when it happens, and many other programs can crash when their attempts to write to disk fail.

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

In rule IDS35-PL. Do not invoke the eval form with a string argument, we replaced the exception **IDS35:EX1** with compliant and noncompliant code examples demonstrating how to load a Perl module specified by a variable. The compliant solution uses `Module::Load`.

Upcoming Events and Training

Training through CERT's
Secure Coding Initiative

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

Our People

Each month we highlight staff members behind our secure coding research. This month we feature Aaron Ballman.

Aaron Ballman has over a decade of experience writing commercial compilers for various languages, and is a Security Software Engineer for CERT. He is an active developer on the clang open source C/C++/Objective-C compiler.

When he's not writing code, Aaron also enjoys being outside, fishing, and reading a good book in his hammock.

Secure Coding Resources

Read David Svoboda's blog post: The CERT Perl Secure Coding Standard

 **Listen** to Developing Secure Software: Universities as Supply Chain Partners

 **Read** The Top 10 Secure Coding Practices

Subscribe to Our eNewsletter

Join the SEI CERT Secure Coding Community