P1706R0: Programming Language Vulnerabilities for C++ update

Date:	2019-06-17 (Pre-Cologne mailing): 10 AM ET
Project:	ISO JTC1/SC22/WG21: Programming Language C++
Audience:	SG12, WG21, WG23
Authors:	Michael Wong (Codeplay)
Contributors:	SG12: Stephen Michel, Peter Sommerlad, Lisa Lippincott, Aaron Ballman, Richard Corden, Clive Pygott, Erhard Ploedereder, John McFarlane, Paul Preney, Andreas Weis, Federico Kircheis, Tullio Vardanega, Jorg Brown,
Emails:	michael@codeplay.com
Reply to:	michael@codeplay.com

Introduction

This document describes the continuing work of reviewing WG23 Programming Vulnerabilities in C++ document in WG21, in SG12.

Revision History

N/A

Motivation and Background

WG23 is an ISO WG under SC22 that looks at programming vulnerabilities of various languages. Since 2017, WG23 has requested a liaison withWG21 to closely collaborate on their work of documenting programming vulnerabilities for C++. This was shown in a presentation [N0729] in the July 2017 WG23 meeting where the background, history of WG23 was presented to SG12. A request was made to WG21 to lead with the documentation of vulnerabilities in C++, so as to ensure their accurate representation using WG21's technical expertise.

This was accepted using a co-located meeting format where WG23 members was to attend WG21 meetings. SG12's charter was expanded from Undefined Behavior and Unspecified Behavior to add Vulnerabilities to the title

SG 12 basically agreed that WG 21 needs to do something about vulnerabilities. It was pointed out that even Ada, widely acknowledged to be a “safe” language, acknowledges 50 of the 63 language-defined vulnerabilities.The goals of this is to work together and not work to make C++ look bad. We are not specifying subsets or what language features to avoid, simply pointing out how vulnerabilities occur and giving guidance. [N0730]

.
As a result these co-located meeting has been occurring in SG12 in every WG21 meeting. This is a report of its progress.

[N0766] describes a Liasion Statement between SC22/WG23 and WG21/SG12.

Liaison Statement

Liaison Statement
SC 22/WG 23 Programming Language Vulnerabilities and
SC 22/WG 21/SG 12 Undefined Behavior and Vulnerabilities and Vulnerabilities Study Group

SC 22/WG 23 and SC 22/WG 21/SG 12 agree to work together to develop TR 24772-9 Programming Language Vulnerabilities in C++.
In developing the Technical Report, the parties agree that the following principles will apply:

A. The TR will, wherever possible, provide meaningful references to existing work (such as the CERT C++ security guidelines and the C++ Core Guidelines)

B. In developing the TR, the parties will evaluate those existing guidelines mentioned in a) (and others) for their safety and security aspects.

C. The parties will help to enhance existing guidelines by alerting the developers of the referenced guidelines of issues identified in those guidelines due to the analysis for the TR.

D. The parties will add new sections to the TR to address new issues found, and to notify developers of other guidelines of such issues.

E. As part of the work, the parties will develop a cross-language taxonomy of issues found from C++ to C and other languages.

F. The parties will explore ways to link the effort described herein with other efforts such as MISRA, AUTOSAR, OpenCL/SYCL SC, and CUDA

G. The work will consider how to provide guidance for previous language versions of C++:2011 and later. Currently identified potential approaches could be

    a. To produce inline guidance for other versions, or

    b. Document guidance for previous version in clause 7 (or even clause 8).

    These will be considered analysis is done clause-by-clause. For example -- strings.

H. The target of the audience is those developing new code vs maintaining old code (in the sense that adding significant functionality under maintenance is also “new code). TR 24772 is generally oriented to the creation of new code, and the coding guidelines for such code. It is expected that old code would only be affected when a major rewrite occurs.

I. The target audience is team leads that produces produce the coding guidelines for the organization, but are experienced C++ programmers, not as opposed to new C++ programmers coming from another language. Goal is not to teach C++.

Impact on the Standard

The result of this is a WG23 document that specifically updates on C++ Programming Vulnerabilities guidelines, along with those from Ada, C, Fortran, etc. These are most of the languages under SC22. We have also been feeding back new findings from this joing WG23/SG12 meeting back to update Core Guidelines.

The goal is to have an active group reviewing various Safety and Vulnerabilities documents and improve them for their accurate portrayals. In future, this will include MISRA C++ as well as AUTOSAR C++ guidelines which is also actively revising their documents. As the various Standards change, we will also need to continue updates. This is still a WIP.

Proposals

Since July 2017, we have held regular meetings. These are the documents that we have generated in WG23. All are located in the WG23 repository:
http://www.open-std.org/JTC1/SC22/WG23/docs/documents

We are about 2/3rd done and anticipate a remaining 35% of work remain for WG23. The latest draft if WG23 can be found here.

http://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0866-tr24772-10-C++-after-mtg-61-20190220.docx

DocumentNumber Description Date

Kona C++ Meeting Feb 2019

N0866 Draft TR 24772-9 C++ Vulnerabilities after Meeting 61 22/02/19

N0860 Draft TR 24772-9 C++ before meeting 61 31/01/19

San Diego C++ meeting 2018

N0841 minutes of meeting 58 08/11/18

N0840 TR 24772-10 C++ language vulnerabilities after meeting 58 09/11/18

N0839 TR 24772-10 C++ language vulnerabilities mid-meeting 58 08/11/18

N0838 TR 24772-10 C++ language vulnerabilities before meeting 58 07/11/18

Rapperswil C++ Meeting June 2018

N0802 TR 24772-10 C++ Language vulnerabilities after meeting 55 08/06/18

N0801 Convenors Report to JTC 1/SC 22 08/06/18

N0800 Minutes of meeting 55 June 6-8 2018
08/06/18

N0799 TR 24772-10 C++ Language vulnerabilities developed during meeting 55 day 1 06/06/18

N0798 Draft Convenors report for 2018 SC 22 plenary 05/06/18

Jackonville C++ meeting March 2018

N0766 Draft liaison statement to WG 21 from pre-meeting 52 telecon 21 Nov 2017

Albuquerque C++ Meeting Nov 2017

N0759 Minutes of meeting 52, held jointly with WG 21/SG 12, November 7-8 2017, Albuquerque, NM 2017-11-08

N0758 TR 24772-9 C++ Vulnerabilities following Meeting 51/WG 21 SG 12 meeting 18/11/17

N0748 C++ Programming language vulnerabilities submitted to WG 21 by Stephen Michell, Michael Wong and Chris Szalwinski 15 October 2017

N0744 Draft of possible C++ Part 9 with updates from Canadian subgroup 11 September 2017

N0741 TR 24772-9 C++ after meeting 50, with sample vulnerability writeup in clause 6.5 17 August 2017

Toronto C++ meeting July 2017

N0730 Report from SC 22/WG 21 C++ meeting held 13 July 2017

N0729 Presentation to SC 22/WG 21 C++ on programming language vulnerabilities 13 July 2017

Acknowledgements

Michael Wong's work is made possible by Codeplay Software Ltd., ISOCPP Foundation, Khronos and the Standards Council of Canada.

References

[N0729]
http://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0729-presentation-WG21-programming-language-vulnerabilities-20170713.ppt
[N0730]
http://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0730-report-from-SC22-WG21-SG12-meeting-20170713.docx
[N0766]
http://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0766-possible-liaison-statement-WG23-WG21-SG12.zip