Business Plan and Convener's Report

ISO/IEC/JTC 1/SC 22/WG 23 (Programming Language Vulnerabilities)

Document: ISO/IEC JTC 1/SC 22/WG 23/N0873

Date: 2020-08-28

PERIOD COVERED: July 2019 – June 2020

SUBMITTED BY:

Convener, ISO/IEC JTC 1/SC 22/WG 23: Vulnerabilities
Stephen Michell
Maurya Software

156 Shinny Ave,
Stittsville, Ontario K2V 0G4 Canada

Office: +1(613)299-9047
E-mail: stephen.michell@maurya.on.ca

1. MANAGEMENT SUMMARY

1.1. JTC 1/SC 22/WG 23 Programming Language Vulnerabilities

1.2. PROJECT REPORT

1.2.1. COMPLETED PROJECTS

ISO/IEC TR 24772-1:2019, *Programing languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 1: Language independent guidance*

Published in December 2019

ISO/IEC TR 24772-2:2020, *Programing languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 2: Vulnerability descriptions for programming language Ada*

Published in January 2020

ISO/IEC TR 24772-3:2020 *Programing languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 3: Vulnerability descriptions for programming language C*

Published in January 2020

The 2012 version of ISO/IEC 24772 has been withdrawn.

ISO/IEC 17960, *Code Signing for Source Code*. This project is to produce an International Standard, and the standard has been published.

1.2.2. PROJECTS UNDERWAY

ISO/IEC TR 24772-4, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 4: Vulnerability descriptions for programming language Python*. This is the update of TR24772:2013 for Python vulnerabilities which was Annex E, following the project split of project 22.24772. Under development

ISO/IEC TR 24772-8, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 8: Vulnerability descriptions for programming language Fortran*. This is the Part for language specific vulnerabilities for Fortran, following the project split of project 22.24772. Under development.

ISO/IEC TR 24772-10, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 10: Vulnerability descriptions for programming language C++*. This is a new Part for language specific vulnerabilities for C++. Under development.

ISO/IEC TR 24772-11, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 11: Vulnerability descriptions for programming language Java*. This is a new Part for language specific vulnerabilities for Java. Being published.

1.2.3. CANCELLED PROJECTS

none

1.2.4. COOPERATION and COMPETITION

Where appropriate, WG 23 has established active liaisons with other SC22

working groups and international organizations, such as Ada Europe and ACM. See the table in 2.3 for a list of liaisons.

There is no apparent direct competition with any other current SC22 working group or JTC 1 subcommittee.

2. PERIOD REVIEW

2.1. MARKET REQUIREMENTS

WG 23 is responding to the needs of the programming language community by inclusion. WG 23 will accept input and liaison by any and all appropriate organizations.

The marketplace demands robust, secure software. Vulnerabilities are the antithesis of robust, secure software. Many of the attacks on software-based systems succeed because the computer language used did not prevent the attack vector and did not warn the developer that the code being produced contained flaws that could be used to generate attacks.

WG 23 has produced 3 editions of TR 24772 (the last one being TR 24772-1:2019, TR 24772-2:2020 and TR 24772-3:2020), but there are vulnerabilities that still need to be identified, and programming languages that still need to be documented with regards to vulnerabilities.

2.2. ACHIEVEMENTS

WG 23 has published the first edition of TR 24772-1, -2 and -3 after splitting the original TR 24772 project and the TR into Part 1, language independent part, and Parts 2, 3, 4, 8, 10 and 11 for language-specific vulnerability descriptions for Ada, C, Python, Fortran, C++ and Java.

2.3. RESOURCES

Seven national bodies have participated in the WG 23 meetings this year: Austria, Canada, China, Italy, Korea, UK, and the USA, as well as several liaisons.

Over the last several years WG 23 has made Web conferencing capabilities available for those that are finding it difficult to travel. At a typical WG 23, one-third to one-half of all participates are remote, but still participate meaningfully in the meeting. WG 23 finds that mixed-mode meetings work well in developing

technical content. WG 23 would like to thank ISO for the Web conferencing support.

Of course, with the world-wide pandemic, WG 23 is holding all meetings virtually.

Liaison with five SC22 Language groups, and four groups outside of SC22 have been established. Liaisons fill a valuable role in that they identify the vulnerabilities that exist (and do not exist) in their language, produce the primary documentation of those vulnerabilities and turn them into the relevant language-dependent part in conjunction with the core team through the liaison individual.

Current WG 23 liaisons are:

Group		Name/Type		Person assigned
SC 22/WG4		Cobol		Robert Karlin,
SC 22/WG5		Fortran		Dan Nagel
SC 22/WG9		Ada		Erhard Ploedereder
SC 22/ WG14		C		Clive Pygott
SC 22/ WG 21		C++		Group
Ada Europe				Erhard Ploedereder
MISRA				Clive Pygott

Ada Europe

3. FOCUS NEXT WORK PERIOD

3.1. DELIVERABLES

WG 23 has the following documents published:

JTC 1 24772-1:2019, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 1: Language Independent Guidance*

JTC 1 24772-2:2020, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 2, Vulnerability descriptions for programming language Ada*

JTC 1 24772-3:2020, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 3, Vulnerability descriptions for programming language C*

WG 23 is working on the following parts:

JTC 1 24772-4, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 4: Vulnerability descriptions for programming language Python*.

JTC 1 24772-8, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 8: Vulnerability descriptions for programming language Fortran*.

JTC 1 24772-10, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 10: Vulnerability descriptions for programming language C++*.

JTC 1 24772-11, *Guidance to Avoiding Vulnerabilities in Programming Languages – Part 11: Vulnerability descriptions for programming language Java*.

3.2. STRATEGIES

WG 23 decided in 2015 that a core document and seven language-specific annexes, with at least two or three more in planning, creates a maintenance burden that makes it difficult to keep all portions of the document up to date in a single document.

WG 23 therefore decided to split TR 24772 into a series of parts, as follows (see also clause 4.1 for the official request for SC 22 action):

- TR24772-1 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 1: Language Independent Guidance*
- TR24772-2 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 2: Ada*
- TR24772-3 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 3: C*
- TR24772-4 *Programming languages — Guidance to avoiding vulnerabilities in programming languages through – Part 4: Python*
- TR24772-5 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 5: Ruby*
- TR24772-6 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 6: SPARK*

- TR24772-7 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 7: PHP*
- TR24772-8 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 8: Fortran*
- TR24772-9 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 9: COBOL*
- TR24772-10 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 10: C++.*
- 24772-11 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 11: Java.*
  This is a new request to SC 22.

3.3. RISKS

Progress on Parts 4, 8, 10, and 11 for which work items are allocated are showing reasonable progress. Editorial and content development meetings are being held bi-weekly for Python, C++ and Java. Some of the other parts for which work items have not been initiated require the identification of resources within other working groups or external experts to undertake the work.

3.4. OPPORTUNITIES

Since the 2019 SC 22 plenary, the US national body has provided resources to develop a Python part, and to develop a Java part.

3.5. WORK PROGRAM PRIORITIES

See 4.1.

4. OTHER ITEMS

4.1. POSSIBLE ACTION REQUESTS AT FORTHCOMING 2020 PLENARY

4.1.1 Free availability of TR 24772-1, -2 and -3

WG 23 requests that SC 22 request free availability of the following documents. The main criteria are that it supports the sale of language standards produced by SC 22 and that its free availability would have little effect on the sale of standards. The two previous editions were approved for free availability, based on the criteria provided at the end of this report, and these documents are un revision of the previous TR 24772.

- ISO/IEC TR 24772-1:2019 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 1: Language Independent Guidance*

- TR24772-2 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 2: Ada*

- TR24772-3 *Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 3: C*

4.1.2 Register the following project with ISO CS

JTC 1 NP TR 24772-11 Programming languages — *Guidance to avoiding vulnerabilities in programming languages – Part 11: Java. (Project Editor Stephen Michell)*

Initiate the following projects with the editors as noted:

- JTC 1 NP TR 24772-4, Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 4: Python. (Project Editor S. Michell)

- JTC 1 NP TR 24772-8, Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 8: Fortran. (Project Editor Dan Nagle)

- JTC 1 NP TR 24772-10, Programming languages — Guidance to avoiding vulnerabilities in programming – Part 10: Programming language C++. (Project Editor Stephen Michell)

- JTC 1 NP TR 24772-11 Programming languages — Guidance to avoiding vulnerabilities in programming languages – Part 11: Java. (Project Editor Stephen Michell)

4.2. ELECTRONIC DOCUMENT DISTRIBUTION

Documents relevant to ISO/IEC/JTC1/SC22 processing are being entered on the ISO eCommittee web site for WG 23. WG 23 conducts some of its detailed technical discussion using the email reflector maintained by Keld Simonsen. WG 23 also has a Web site at http://open-std.org/jtc1/sc22/wg23.

4.4. RECENT MEETINGS

No	Date	Place	# attendees	Host
57	27-28 Aug 2018	Zoom	6	Convenor

58	8-9 Nov 2018	San Diego CA with WG 21	9	USA, WG 21
59	21 January 2019	Zoom Meeting	5	N/A
60	Cancelled			
61	20-22 Feb 2019	Kona, Hawaii with WG 21	7	USA, WG 21
62	6 May 2019	Zoom Meeting	5	N/A
63	16-18 July 2019	Cologne Germany with WG 21		Germany, WG 21
64	Cancelled			
65	15 October 2019	Zoom Meeting	6	N/A
66	6-9 November 2020	Belfast, UK with WG 21	6	UK, WG 21
67	10-12 Feb 2020	Prague, Czech Republic with WG 21	15	Czech, WG 21
68	23-24 Feb 2020	Las Vegas NV with INCITS Fortran	6	US, INCITS
69	19 May 2020	Zoom Meeting	5	Convenor

In addition, more than 10 meetings have been held with dedicated language experts to progress the development of Part 10 C++, Part 4 Python and Part 11 Java.

4.5. FUTURE MEETINGS

#70 Zoom Meeting Sep 14,15 2000-2200 UTC
#71 USA or Zoom meeting Nov 2020 (with WG 21)

#72 Kona, HI or Zoom	22 Feb 2021
#73 Las Vegas with WG 5	TBD June 2020

(5) REFERENCE MODELS A:
Standards which explain the relationships between existing standards

Justification Ease of consensus

Catalogues of standards for sales ++ promotion

The JTC 1/SC 22 secretariat requests that the JTC 1 secretariat take the necessary action to make ISO/IEC TR 24772, *Information Technology — Programming Languages — Guidance to Avoiding Vulnerabilities in Programming Languages through Language Selection and Use,* publicly available and free of charge.

ISO/IEC TR 24772 describes security and safety vulnerabilities that can arise from the undisciplined use of programming languages, including languages maintained by ISO/IEC JTC 1/SC 22. It also describes how improved use of the languages allows one to avoid the vulnerabilities. The free availability of 24772 would promote the use of JTC 1 programming languages by demonstrating how they can be used in a safe and secure manner.

N07604

All of the JTC 1 programming languages were developed in an era prior to the ubiquitous connectivity of today's computers. Their designers paid little attention to the problems of "hacking" by unauthorized users. Therefore these languages contain features that, when improperly used, make the program vulnerable to attack from unauthorized users. Language developers and maintainers, including SC 22 working groups, have paid increasing attention to the problem in recent years and now provide alternative features or alternative ways to use existing features that mitigate the problem. Unfortunately, this is not well known. For example, the C language is commonly accused of having a weakness in its facility for string copying, despite the fact that the standard now provides an alternative library that does not have the weakness.

The purpose of TR 24772 is to survey the subject of vulnerabilities in programming languages and to provide generic descriptions of the vulnerabilities and the ways to mitigate them. The first edition of the report is completely language-independent. Future editions will contain annexes for individual programming languages relating the language-independent descriptions to the specific features of the specific language. The TR can play an important role in bolstering confidence in the SC 22 programming languages.

Therefore, with respect to the criteria cited above:

(5) The language-specific annexes of TR 24772 will call out many of the language standards of SC22. Existing freely available material[2] on similar subjects has the effect of directing persons away from the ISO programming languages. Our material will have the effect of directing users toward the standardized languages because we emphasize adherence to the ISO standards as the most basic step to address the problem.

(6) TR 24772 includes recommendations to the architects of programming languages regarding areas that they might address in future revisions. It demonstrates the commitment of JTC 1 to meet the challenges of modern Information Technology. TR 24772 does not contain normative provisions.

(8) TR 24772 explains how to use standard ISO programming languages in manners that are appropriate to the modern challenges of computing security and safety. The Technical Report makes direct references to the ISO language standards.

In this particular case, it is also useful to describe the situation with respect to the "rules for selection of the criteria," also listed in N7269:

Rules for selection of the criteria	**Comments regarding TR 24772**
(1) Insignificant impact on revenue by free access	TR 24772 cannot be used as a substitute for any of the SC 22 standards. It does not even provide summaries of them.
(2) Promotion of the sales of other JTC 1 documents	TR 24772 helps to improve public awareness of JTC 1 programming languages, the importance of using the standard languages, and the steps that have been taken to improve the standards.
(3) Enhancement of awareness and dominance of JTC 1 work	TR 24772 demonstrates that JTC 1 is the best and most responsible venue for programming language specification