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/N1321

Date: 2022-09-07

PERIOD COVERED: July 2021 – September 2022

SUBMITTED BY:

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

38 D'Arcys Way,
Kemptville, Ontario K0G 1J0 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 The 2012 version of ISO/IEC 24772 has been withdrawn.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 DIS 24772-1, *Programing languages – Avoiding Vulnerabilities in Programming Languages – Part 1: Language independent catalogue of vulnerabilities.* This is a rework of TR 24772-1:2019 to make it an international standard. Document has passed DIS ballot, comments have been integrated and the document is almost ready for FDIS ballot.

ISO/IEC WD 24772-2, *Programing languages – Avoiding Vulnerabilities in Programming Languages – Part 2: Vulnerability descriptions for the programming language Ada*. This is a rework of TR 24772-2:2020 to make it an international standard.

ISO/IEC WD 24772-3:2020 *Programing languages – Avoiding Vulnerabilities in Programming Languages – Part 3: Vulnerability descriptions for the programming language C.* This is a rework of TR 24772-3:2020 to make it an international standard.

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

ISO/IEC WD 24772-6, *Programing languages – Avoiding Vulnerabilities in Programming Languages – Part 6: Vulnerability descriptions for programming language SPARK.* This is a complete rewrite of ISO/IEC TR 24772:2013 for SPARK vulnerabilities which was Annex H, following the project split of project 22.24772. Significant changes to the SPARK language necessitated a major rewrite. WD development is complete.

ISO/IEC WD 24772-8, *Programing languages – 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. Document is under final editing before submission for DIS ballot.

ISO/IEC WD 24772-10, *Programing languages – 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 WD 24772-11, *Programing languages – Avoiding Vulnerabilities in Programming Languages – Part 11: Vulnerability descriptions for programming language Java.* This is a new Part for language specific vulnerabilities for Java. Under development. The Java community, led by Oracle, has objected to the creation of a document that exposes Jave vulnerabilities. Recent discussions with Oracle have changed their position and they are ready to review the document for completeness and accuracy.

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.

In addition, ISO and IEC have changed the requirements for a Technical Report. The 2021 Directives state that new technical reports can no longer provide guidance nor requirements. WG 23 has therefore in the position where it must make the 24772 series into international standards.

At the same time, we are hoping that we will be able to make these documents freely available. Part of the name change is to be clear that the 24772 series catalogues vulnerabilities and documents avoidance mechanisms.

WG 23 also decided that the standards that document programming languages should not contain requirements on developers. The reasons for this decision stem from the wide variety of safety and security levels that programming languages and the users of these languages must address. What could be a requirement for one program could be optional for an organization meeting a lower safety or security requirement. Hence 24772 series documents can only recommend techniques, rules and avoidance mechanisms to organizations and users.

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

Six national bodies have participated in the WG 23 meetings this year:, Canada, Italy, Spain, Switzerland, 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 face-to-face 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. With the world-wide pandemic, WG 23 is holding all meetings virtually but may move back to in-person meetings for meetings that require larger group discussions.

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		Steve Lionel
SC 22/WG9		Ada		Erhard Ploedereder
SC 22/ WG14		C		Clive Pygott
SC 22/ WG 21		C++		Stephen Michell
Ada Europe				Erhard Ploedereder
MISRA				Clive Pygott

3. FOCUS NEXT WORK PERIOD

3.1. DELIVERABLES

WG 23 has the following documents published:

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

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

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

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

WG 23 is working on the following parts of 24772:

- ISO/IEC DIS 24772-1, *Programming languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 1: Language Independent Guidance*
- ISO/IEC WD 24772-2, *Programming languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 2, Vulnerability descriptions for programming language Ada*
- ISO/IEC WD 24772-3, *Programming languages – Guidance to Avoiding Vulnerabilities in Programming Languages – Part 3, Vulnerability descriptions for programming language C*
- ISO/IEC 24772-4, *Programming languages – Avoiding Vulnerabilities in Programming Languages – Part 4: Vulnerability descriptions for programming language Python.*
- ISO/IEC 24772-8, *Programming languages –Avoiding Vulnerabilities in Programming Languages – Part 8: Vulnerability descriptions for programming language Fortran.*
- ISO/IEC 24772-10, *Programming languages – Avoiding Vulnerabilities in Programming Languages – Part 10: Vulnerability descriptions for programming language C++.*
- ISO/IEC 24772-11, *Programming languages – 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 recommended and SC 22 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.

Once TR 24772-1, TR 24772-2 and TR 24772-3 were published, ISO refused free availability for the technical reports. Then for 2021, ISO refuses to publish technical reports that contain guidance, which is what all these documents provide. Hence, WG 23 wishes to publish all of these documents as freely available international standards. At the time of this report, WG 23 is submitting 24772-1 to SC 22 for registration as a NP and for an immediate DIS ballot.

Within the next 4 months, WG 23 expects to submit the following documents for NWIP ballot and simultaneous DIS ballot:

- 24772-2 *Programming languages – Avoiding vulnerabilities in programming languages – Part 2: Vulnerability descriptions for programming language Ada*
- 24772-2 *Programming languages – Avoiding vulnerabilities in programming languages – Part 3: Vulnerability descriptions for programming language C*
- 24772-2 *Programming languages – Avoiding vulnerabilities in programming languages – Part 6: Vulnerability descriptions for programming language SPARK*
- 24772-2 *Programming languages – Avoiding vulnerabilities in programming languages – Part 4: Vulnerability descriptions for programming language Python*

3.3. RISKS

Progress on Parts 1, 2, 3, 4, 6, 8, 10, and 11 for which work items are allocated are showing

reasonable progress. Editorial and content development meetings are being held bi-weekly or tri-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 2022 PLENARY

- Renew the convenor, Stephen Michell for another three (3) years, or name a new convenor.
- Ensure that 24772-1 is submitted for FDIS ballot.
- Pursue free availability for 24772-1
- Ensure that project titles and document titles are as documented in this report.

4.2. ELECTRONIC DOCUMENT DISTRIBUTION

Documents relevant to ISO/IEC/JTC1/SC22 processing will be 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
72	22 Feb 2021	Zoom Meeting	5	Convenor
73	21 Jul 2021	Zoom Meeting	5	Convenor

74	21 Jan 2022	Zoom Meeting	5	Convenor
75	12 Apr-10 May 2023	Zoom Meeting	8	Convenor

In addition, more than 50 subgroup meetings (on average weekly with one or two language groups meeting every third or fourth week) have been held with dedicated language experts to progress the development of Part 10 C++, Part 4 Python, and Part 8 Fortran.

4.5. FUTURE MEETINGS

Due to the pandemic, all WG 23 meetings are being held virtually in small language-specific groups ranging from 5 to 12 individuals. When a general topic arises that needs a formal decision, a targeted meeting is called.

Once the pandemic is declared over and ISO/IEC permit face-to-face meetings we will resume having formal in-person WG meetings, if the convenor’s family situation permits.