ISO/IEC #####-#:####(X)

ISO/IEC JTC 1/SC 22/WG 23

Secretariat: ANSI

Information Technology — Programming Languages — Top Software Vulnerability Mitigations

WD/CD/DIS/FDIS stage

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Foreword iv

Introduction v

1 Scope 1

2 Conformance 1

3 Normative references 1

4 Terms and definitions 1

5 Requirements 3

Bibliography 5

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology, Subcommittee* SC22, *Programming languages, their environments and system software interfaces*.

A list of all parts in the ISO 24772 series can be found on the ISO website.

Introduction

All programming languages contain constructs that are incompletely specified, exhibit undefined behaviour, are implementation-dependent, or are difficult to use correctly. The use of those constructs may therefore give rise to vulnerabilities, as a result of which, software programs can execute differently than intended by the writer. Software design can introduce software vulnerabilities as well. Insufficient checking of input, or incorrect assignment or poor tracking of privileges can easily introduce weaknesses in the software. In some cases, these weaknesses can be exploited to compromise the safety of a system or be exploited by attackers to compromise the security or privacy of a system.

This International Standard is intended to provide a common set of core principles spanning multiple programming languages, so that application developers can avoid the software issues that lead to vulnerabilities in software written in their chosen software language. It should be noted that this standard establishes a minimum set of core principles and thus is inherently incomplete. The standard is purposely minimized for several reasons. The standard must apply to the general case. It must cover the wide swath of computer programming languages and application needs. The minimum set is just that – a minimum. It is expected that this will augmented with additional requirements by the users of this standard, and serve as a basis for other standards or certifications that offer more stringent requirements needed for particular applications.

Information Technology — Programming Languages — Top Software Vulnerability Mitigations

1 Scope

This document specifies software programming language vulnerabilities to be avoided in the development of systems where assured behaviour is required for security, safety, mission-critical and business-critical software. In general, this standard is applicable to the software developed for any application.

2 Conformance

An implementation of **software** conforms to this International Standard if it meets the requirements specified in Clause **5**.

3 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 24772:2010 Information technology -- Programming languages -- Guidance to avoiding vulnerabilities in programming languages through language selection and use

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382 and the following apply.

***all of the following are from ISO/IEC 2382 except “control sphere”, can I still list them for convenience?

3.1
access control

means of ensuring that the resources of a data processing system can be accessed only by authorized entities in authorized ways

3.2

assignment statement/assignment

simple statement that replaces the current data value of a variable with a new data value specified by an expression

3.3

authentication

<security> act of verifying the claimed identity of an entity

3.4

authorization

granting of rights, which includes the granting of access based on access rights

3.5

control sphere

set of resources and behaviors that are accessible to a single actor, or a group of actors that all share the same security restrictions

3.6

credentials

data that are transferred to establish the claimed identity of an entity

3.7

dynamic

pertaining to a data attribute, whose values can only be established during the execution of all or part of a program

3.8

error

discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition

3.9

error correction

method used to correct erroneous data produced during data transmission, transfer, or storage

3.10

error detection

method of determining whether data has been transmitted or transferred incorrectly

3.11

error recovery

process of correcting or bypassing the effect of a fault or an error to allow the functional unit to continue to perform a required function

3.12
input data

data entered into an information processing system or any of its parts for storage or processing

3.13

password

character string that is used as authentication information

3.14

scope/scope of a declaration

that portion of a program within which a declaration is valid

3.15
security vulnerability

weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat

3.16

sensitive information

information that, as determined by a competent authority, must be protected because its disclosure, modification, destruction, or loss will cause perceivable damage to someone or something

3.17

source code

code expressed in a form suitable for input to an assembler, compiler, or other translator

3.18

static

pertaining to objects that exist and retain their values throughout the execution of the entire program

5 Requirements

The list of requirements described below is intended to be language and platform independent.

Conformance to this International Standard requires that:

1. Input data shall be validated for type, length, format, and range, known valid and safe data.

2. Compiler static analysis checking shall be enabled and any resultant security issues shall be resolved.

3. The source code shall be run through a static source code analysis tool, in addition to undergoing compiler static analysis, to detect security anomalies and any resultant security anomalies shall be resolved.

4. Explicit bounds checking shall be performed at the point of use when it cannot be shown statically that bounds will be obeyed, when bounds checking is not provided by the implementation, or if automatic bounds checking is disabled.

5. Dynamic memory, files, tasks and threads shall be allocated and freed within the same scope.

6. Error detection, error reporting, error correction, and error recovery shall be implemented at instances where error conditions could occur.

7. Deprecated language features shall not be used.

8. Sensitive data shall be sanitized, erased or encrypted to prevent it from being visible to others such as when it is in freed memory or in transmitted data.

9. Default credentials, such as passwords, shall be required to be changed upon first use.

10. All code shall be attributed to a responsible party through the use of configuration management or some other tracking mechanism.

11. Authentication and authorization, including checking for expired or revoked credentials, shall be performed before conducting a privileged operation.

12. Code shall be reviewed and tested to detect incorrect coding, control flow and comparisons.

13. Reasonable limits shall be placed on the number of resources that can be requested.

14. Proper synchronization/locking shall be used when accessing shared resources to prevent race conditions or deadlock.

15. Values, variables and resources shall not be used outside of their scope and lifetime.

Bibliography

[1] *Common Weakness Enumeration (CWE) Glossary,* *https://cwe.mitre.org/documents/glossary/index.html*

[2] ISO/IEC 24772:2010 *Information technology -- Programming languages -- Guidance to avoiding vulnerabilities in programming languages through language selection and use*