International Standard

ISO/IEC 18974

First edition
2023-12

Information technology — OpenChain security assurance specification

Reference number
ISO/IEC 18974:2023(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of 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 www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Joint Development Foundation (JDF) (as OpenChain Security Assurance Specification 1.1) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The OpenChain Project (see [4]) is working towards a supply chain where open source is delivered with trusted and consistent compliance information. As part of this mission, the OpenChain Project maintains ISO/IEC 5230 (see [1]), the International Standard for open source license compliance. A natural next step in support of the broader mission was to develop a guide to identify and present the minimum core set of requirements every security assurance program should satisfy with respect to the use of open source software.

For context, ISO/IEC 5230 is a process management specification that identifies inbound, internal and outbound inflection points where a process, policy or training should exist. The identification and tracking of software used and deployed is an inherent part of getting this right, and this allows the approach to also be useful for security or export control.

The OpenChain Project community noticed ISO/IEC 5230 being used in the security domain and decided to develop this security specification to satisfy market demand. This specification is intended to identify and describe the key requirements of a quality security assurance program in the context of using open source Software. It focuses on a narrow subset of primary concern: checking open source Software against publicly known security vulnerabilities like CVEs, GitHub/GitLab vulnerability reports, and so on.

This specification focuses on the "what" and "why" aspects of a quality security assurance program rather than delving into to "how" and "when." This was a conscious decision to ensure flexibility for organizations of any size and in any market to use this specification. This approach, along with the types of processes identified, is built on more than five years of practical, global feedback around the creation and management of such programs. The result is that a company can frame a program that precisely fits their supply chain requirements, scoped to a single product or a complete legal entity, and take this solution to market quickly and effectively.

This specification was derived from [5]. That reference document went through a final approval process via the OpenChain Project's normal voting practice to transform into this published security specification. The scope of this specification may expand over time based on community feedback.

Clause 4 defines the requirements that a program must satisfy to achieve a core level of security assurance. Each requirement consists of one or more verification materials (i.e., records) that must be produced to satisfy the requirement. Verification materials are not required to be made public, though an organization may choose to provide them to others, potentially under a Non-Disclosure Agreement (NDA).

This specification is maintained by the OpenChain Project. Information about participation in that maintenance is available at https://www.openchainproject.org/community.

Information technology — OpenChain security assurance specification

1 Scope

This specification contains the key requirements of a quality open source software security assurance program that establishes trust between organizations exchanging software solutions comprised of open source software.

2 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.

Free Software Definition, Free Software Foundation, www.gnu.org/philosophy/free-sw.html

The Minimum Elements For a Software Bill of Materials (SBOM), The United States Department of Commerce, July 12, 2021, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

Open Source Definition, Open Source Software Initiative, www.opensource.org/osd

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
component record
data set containing supplier name, component name, version of the component, other unique identifiers, dependency relationship, author of SBOM data and timestamp, as defined by NTIA minimum elements for SBoM

3.2
customer agreement
agreement that the processes used to find, track or fix security issues are regarded as sufficient and correct by relevant customers or user organizations if applicable

3.3
common vulnerabilities and exposures
CVE
disclosed computer software security issues and flaws in a public database

Note 1 to entry: When someone refers to a CVE, they mean a security flaw that has been assigned a CVE ID number within the database. The CVE database is sponsored by the US Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA).

3.4
documented evidence
explicitly stored data that outlines, explains or records information related to activities and actions referred to in this specification

3.5
known vulnerability
security vulnerability previously discovered in *open source software* (3.7) components that are publicly available, including any publicly published vulnerabilities including, but not limited to, CVEs, GitHub/GitLab vulnerability alerts, and package manager alerts

3.6
newly discovered vulnerability
security vulnerability just discovered in *open source software* (3.7) components that are publicly available, including any publicly published vulnerabilities including, but not limited to, *CVEs* (3.3), GitHub/GitLab vulnerability alerts, and package manager alerts

3.7
open source software
software subject to one or more licenses that meet the Open Source Definition published by the Open Source Software Initiative or the Free Software Definition published by the Free Software Foundation

3.8
program
the set of policies, processes and personnel that comprise an organization's security assurance activities

3.9
program participant
any employee or contractor that defines, contributes to or has responsibility for preparing supplied software

Note 1 to entry: Depending on the organization, that may include, but is not limited to, a software developer, a release engineer, a quality engineer, or someone in product marketing, product management or procurement.

3.10
security assurance
the confidence that a system meets the requirements for security best practices and is resilient against *known vulnerabilities* (3.5)

3.11
security testing
process for the analysis of software (or other components) that allows for understanding their current and potential future management in the context of *known vulnerabilities* (3.5)

3.12
software bill of materials
SBOM
information in a structured format such as SPDX (see [2]) that allows the exchange of information for a software package, which might usefully include name, version, origin, license, copyright and *known vulnerabilities* (3.5) in a manner useful to third parties

3.13
supplied software
software that an organization distributes or makes available to third parties (e.g., other organizations or individuals)

3.14
verification materials
materials that demonstrate that a given requirement of the specification is satisfied

4 Requirements

4.1 Program foundation

4.1.1 Policy

A written policy shall be created that governs open source software security assurance of supplied software. The policy shall be internally communicated. The policy and its method of communication shall have a review process to ensure they are current and relevant.

Verification material(s):

— 4.1.1.1: A documented open source software security assurance policy;

— 4.1.1.2: A documented procedure to make program participants aware of the security assurance policy.

Rationale:

This is to make sure a process exists to create, record, and make program participants aware of the existence of an open source software security assurance policy. Although no requirements are provided here on what should be included in the policy, other sections may impose additional requirements.

4.1.2 Competence

The organization shall:

— Identify the roles and responsibilities that impact the performance and effectiveness of the program;

— Determine the necessary competence of program participants fulfilling each role;

— Ensure that program participants have appropriate education, training, and/or experience;

— Where applicable, ensure program participants take actions to acquire the necessary competence;

— Retain appropriate documented information as evidence of competence as well as who is currently a participant in the program.

Verification material(s):

— 4.1.2.1: A documented list of roles with corresponding responsibilities for the different program participants;

— 4.1.2.2: A document that identifies the competencies for each role;

— 4.1.2.3: List of participants and their roles;

— 4.1.2.4: Documented evidence of assessed competence for each program participant;

— 4.1.2.5: Documented evidence of periodic reviews and changes made to the process;

— 4.1.2.6: Documented verification that these processes are current with company internal best practices and who is assigned to accomplish them.

Rationale:

To ensure that program participants have a sufficient level of competence for their respective roles and responsibilities.

4.1.3 Awareness

The organization shall ensure that the program participants are aware of:

— The open source software security assurance policy;

— Relevant program objectives;

— Their contribution to the effectiveness of the program;

— The implications of not following the program's requirements.

Verification material(s):

— 4.1.3.1: Documented evidence of assessed awareness for the program participants - which should include the program's objectives, one's contribution within the program, and implications of program non-conformance.

Rationale:

To ensure the program participants have obtained a sufficient level of awareness for their respective roles and responsibilities within the program.

4.1.4 Program scope

A program should have defined guiding principles and scope that match the risk management policy of the entire organization. It should be clear whether the program applies to a product line, a department, or the entire organization. It should also be understood that this scope may change over time and metrics may be used to assess its ongoing effectiveness.

Verification material(s):

— 4.1.4.1: A written statement that clearly defines the scope and limits of the program;

— 4.1.4.2: A set of metrics the program shall achieve to improve;

— 4.1.4.3: Documented evidence from each review, update, or audit to demonstrate continuous improvement.

Rationale:

To provide the flexibility to construct a program that best fits the scope of an organization's needs. Some organizations could choose to maintain a program for a specific product line while others could implement a program to govern the supplied software of the entire organization.

4.1.5 Standard practice implementation

The program shall demonstrate sound and robust handling procedures of known vulnerabilities and secure software development by defining and implementing following procedures:

— Method to identify structural and technical threats to the supplied software;

— Method for detecting existence of known vulnerabilities in supplied software;

— Method for following up on identified known vulnerabilities;

— Method to communicate identified known vulnerabilities to customer base when warranted;

— Method for analyzing supplied software for newly published known vulnerabilities post release of the supplied software;

— Method for continuous and repeated security testing to be applied for all supplied software before release;

— Method to verify that identified risks will have been addressed before release of supplied software;

— Method to export information about identified risks to third parties as appropriate.

A process shall exist for the security assurance methods listed above.

Verification material(s):

— 4.1.5.1: A documented procedure exists for each of the methods identified above.

Rationale:

To ensure appropriate processes exists for detecting and following up on known vulnerabilities in the supplied software.

4.2 Relevant tasks defined and supported

4.2.1 Access

The program shall maintain a process to effectively respond to external known vulnerability inquiries. To accomplish this, it shall publicly identify a means for third parties to inquire about how a known vulnerability impacts a software offering.

Verification material(s):

— 4.2.1.1: Publicly visible method to allow third parties to make known vulnerability or newly discovered vulnerability enquires (e.g., via an email address or web portal that is monitored by program participants);

— 4.2.1.2: An internal documented procedure for responding to third party known vulnerability or newly discovered vulnerability inquiries.

Rationale:

To ensure there is a reasonable way for third parties to contact securely the organization regarding security vulnerability inquiries and that the organization is prepared to respond.

4.2.2 Effectively resourced

Ensure the following tasks relevant to the program are identified and resourced to meet the process requirements of this document:

— Assign accountable personnel to ensure the successful execution of program tasks;

— Program tasks are sufficiently resourced;

— Sufficient time to perform the tasks have been allocated;

— Adequate funding has been allocated;

— A process exists for reviewing and updating the policy and supporting tasks;

— Technical expertise pertaining to known vulnerabilities is accessible to those who may need such guidance.

Verification material(s):

— 4.2.2.1: Document with name of persons, group or function in program role(s) identified;

— 4.2.2.2: The identified program roles have been properly staffed and adequate funding provided;

— 4.2.2.3: Identification of expertise available to address identified known vulnerabilities;

— 4.2.2.4: A documented procedure that assigns internal responsibilities for security assurance.

Rationale:

To ensure: i) program responsibilities are effectively supported and resourced and ii) policies and supporting processes are regularly updated to accommodate changes in security assurance best practices.

4.3 Open source software content review and approval

4.3.1 Software bill of materials

A process shall exist for creating and maintaining a bill of materials that includes each open source software component from which the supplied software is comprised.

Verification material(s):

— 4.3.1.1: A documented procedure ensuring all open source software used in the supplied software is continuously recorded across the lifecycle of the supplied software. This includes an archive of all open source software used in the supplied software;

— 4.3.1.2: open source software component records for the supplied software that demonstrates the documented procedure was properly followed.

Rationale:

To ensure a process exists for creating and managing a software bill of materials used to construct the supplied software. A bill of materials is needed to support the systematic review of each component to understand if any known vulnerabilities exist.

4.3.2 Security assurance

A process shall exist to ensure each open source software component to be included in the software bill of materials for the supplied software will have some security assurance activities applied.

— Apply method for detecting existence of known vulnerabilities;

— For each identified known vulnerability assign a risk/impact score;

— For each detection and assigned score determine and document necessary remediation or mitigation steps suitable for the use-case of the software;

— Obtain Customer Agreement that the proposed resolution is acceptable if necessary; at or above a previously determined level (i.e., all severity scores above 4.5 …);

— Depending on the risk/impact score take the appropriate action (e.g., contact customers if necessary, upgrade software component, no further action, …);

— If a newly discovered vulnerability is present in previously distributed supplied software, depending on the risk/impact score take the appropriate action (e.g., contact customers if warranted);

— An ability to monitor supplied software after its release to market and to respond to known vulnerability or newly discovered vulnerability disclosures.

Verification material(s):

— 4.3.2.1: A documented procedure for handling detection and resolution of known vulnerabilities for the open source software components of the supplied software;

— 4.3.2.2: For each open source software component a record is maintained of the identified known vulnerabilities and action(s) taken (including even if no action was required).

Rationale:

To ensure the program is sufficiently robust to handle the identified known vulnerabilities for the open source software from which the supplied software is comprised. That a procedure exists to support this activity and that the procedure is followed.