OpenChain Security Assurance Reference Guide 0.1

Contents

Introduction iii
1 Scope 1
2 Terms and definitions 1
3 Requirements 2
3.1 It is recommended that you are OpenChain ISO 5230 conforming 2
3.2 Program foundation 2
3.2.1 Policy 2
3.2.2 Competence 2
3.2.3 Awareness 2
3.2.4 Program scope 3
3.2.5 Standard Practice Implementation 3
3.3 Relevant tasks defined and supported 3
3.3.1 Access 3
3.3.2 Effectively resourced 4
3.4 Open source content review and approval 4
3.4.1 Bill of materials 4
3.4.2 Security Assurance 5
~~3.5 Compliance artifact creation and delivery~~ 5
3.5.1 ~~Security Assurance Artifacts~~ 5
3.6 Adherence to the guideline requirements 5
3.6.1 Assurance 5
3.6.2 Duration 6

Introduction

The OpenChain Specification working group's core mission is to develop program standards that establish trust in the open source from which modern-day software solutions are built. The OpenChain project's flagship specification, ISO 5230 International Standard, is currently focused on establishing trust around open source license compliance. In support of its mission, the working group has embarked on an exploratory initiative to develop a set of requirements that serve as a benchmark every quality open-source security assurance program would satisfy.

Conformance with this reference guide provides assurance that an organization has a program in place that takes the expected steps necessary to establish a high level of security assurance with respect to the open source used. This document focuses on the "what" and "why" aspects of a program rather than the "how" and "when". This ensures flexibility for different organizations of different sizes in different markets to choose specific policy and process content that fits their size, goals and scope. For instance, a conformant program may address a single product line or the entire organization.

This introduction provides the context for all potential users. Section 2 defines key terms used throughout this document. Section 3 defines the requirements that a program must satisfy to achieve a sufficient 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).

OpenChain Security Assurance Reference Guide

1 Scope

This document specifies the key requirements of a quality open source security assurance program in order to provide a benchmark that builds trust between organizations exchanging software solutions comprised of open source software.

2 Terms and definitions

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

2.2 known vulnerabilities
Common Vulnerabilities and Exposures (CVE) is a public database of disclosed computer software security issues and flaws. When someone refers to a CVE, they mean a security flaw that's been assigned a CVE ID number within the database. CVE database is sponsored by the US Federal Government, with both the US Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA) contributing operating funds.

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

2.2 program participants
any organization employee or contractor that defines, contributes to or has responsibility for preparing supplied software

Note: Depending on the organization, that may include (but is not limited to) software developers, release engineers, quality engineers, product marketing and product management.

2.3 security assurance
the confidence that a system meets the requirements for security best practices and is resilient against publicly disclosed computer security vulnerabilities and flaws.

2.4 SPDX
the format standard created by the Linux Foundation's SPDX (Software Package Data Exchange) Working Group for exchanging bill of materials for a given software package, including associated license and copyright information (see spdx.org)

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

2.6 verification materials
materials that demonstrate that a given requirement of the reference guide is satisfied

3 Requirements

3.1 It is recommended that you are OpenChain ISO 5230 conforming

3.2 Program foundation

3.2.1 Policy

A written policy shall exist that governs open source security assurance of the supplied software. The policy shall be internally communicated.

Verification material(s):

- 3.1.1.1 A documented open source security assurance policy.
- 3.1.1.2 A documented procedure that makes program participants aware of the existence of the secure assurance policy (e.g., via training, internal wiki, or other practical communication method).

Rationale:

To ensure steps are taken to create, record and make program participants aware of the existence of an open source secure assurance policy. Although no requirements are provided here on what should be included in the policy, other sections may impose requirements on the policy.

3.2.2 Competence

The organization shall

- Identify the roles and the corresponding responsibilities of those roles that affects the performance and effectiveness of the program;
- Determine the necessary competence of program participants fulfilling each role
- Ensure that program participants are competent on the basis of appropriate education, training, and/or experience;
- Where applicable, take actions to acquire the necessary competence; and
- Retain appropriate documented information as evidence of competence.

Verification material(s):

- 3.1.2.1 A documented list of roles with corresponding responsibilities for the different participants in the program.
- 3.1.2.2 A document that identifies the competencies for each role.
- 3.1.2.3 Documented evidence of assessed competence for each program participant.

Rationale:

Ensure that the program participants have obtained a sufficient level of competence for their respective roles and responsibilities.

3.2.3 Awareness

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

- The open source security assurance policy;
- Relevant open source objectives;
- Their contribution to the effectiveness of the program; and
- The implications of not following the Program's requirements.

Verification material(s):

- 3.1.3.1 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.

3.2.4 Program scope

Different programs may be governed by different levels of scope. For example, a program could govern a single product line, an entire department or an entire organization. The scope designation needs to be declared for each program.

Verification material(s):

- 3.1.4.1 A written statement that clearly defines the scope and limits of the program.

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.

3.2.5 Standard Practice Implementation

- Organization Knowledge of CVEs
- Method for detecting existence of CVEs
- Method for following up of identified CVEs
- Method to Communicate security vulnerabilities to customer base when warranted
- Method for Analyzing post distribution

A process shall exist with for common standard secure assurance methods.

Verification material(s):

- 3.1.5.1 A documented procedure on how an organization detects and follows up on known vulnerabilities.

Rationale:

To ensure a process exists for detecting and following up on known security vulnerabilities.

3.3 Relevant tasks defined and supported

3.3.1 Access

Maintain a process to effectively respond to external security vulnerability inquiries. Publicly identify a means by which a third party can inquire about a known CVE with respect to a given software offering.

Verification material(s):

- 3.2.1.1 Publicly visible method that allows any third party to make a security vulnerability inquiry (e.g., via a published contact email address, or the Linux Foundation's Open Compliance Directory).

- ☐ 3.2.1.2 An internal documented procedure for responding to third party security vulnerability inquiries exists.

Rationale:

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

3.3.2 Effectively resourced

Identify and Resource Program Task(s):

- Assign accountability to ensure the successful execution of program tasks.
- Program tasks are sufficiently resourced:
  - Time to perform the tasks have been allocated; and
  - Adequate funding has been allocated.
- A process exists for reviewing and updating the policy and supporting tasks; and
- Technical expertise pertaining to security vulnerabilities is accessible to those who may need such guidance;

Verification material(s):

- ☐ 3.2.2.1 Document with name of persons, group or function in program role(s) identified.
- ☐ 3.2.2.2 The identified program roles have been properly staffed and adequate funding provided.
- ☐ 3.2.2.3 Identification of security vulnerability expertise available to address CVE considerations which could be internal or external.
- ☐ 3.2.2.4 A documented procedure that assigns internal responsibilities for security assurance.
- ☐ 3.2.2.5 A documented procedure for handling the review and remediation of identified security cases.

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.

3.4 Open source content review and approval

3.4.1 Bill of materials

A process shall exist for creating and managing a bill of materials that includes each open source component (and its identified known vulnerabilities) from which the supplied software is comprised.

Verification material(s):

- ☐ 3.3.1.1 A documented procedure for identifying, tracking, reviewing, approving, and archiving information about the collection of open source components from which the supplied software is comprised.
- ☐ 3.3.1.2 Open source component records for the supplied software that demonstrates the documented procedure was properly followed.

Rationale:

To ensure a process exists for creating and managing an open source component 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 security vulnerabilities exist

3.4.2 Security Assurance

- For each open source component in the Bill of Materials
  - Apply method for detecting existence of known security vulnerabilities
  - For each identified known security vulnerability assign a risk/impact score
  - Depending on the risk/impact score take the appropriate action (e.g., contact customers if warranted)
  - If present in previously distributed software solutions, depending on the risk/impact score take the appropriate action (e.g., contact customers if warranted)

Verification material(s):

- 3.3.2.1 A documented procedure for handling detection and resolution of known security vulnerabilities for the open source components of the supplied software.

Rationale:

To ensure the program is sufficiently robust to handle an organization's security assurance for the open source from which their software solutions are comprised. That a procedure exists to support this activity and that the procedure is followed.

~~3.5 Compliance artifact creation and delivery~~

~~3.5.1 Security Assurance Artifacts~~

~~A process shall exist for creating the set of compliance artifacts for the supplied software.~~

~~**Verification material(s):**~~

- ~~3.4.1.1 A documented procedure that describes the process under which the compliance artifacts are prepared and distributed with the supplied software as required by the identified licenses.~~
- ~~3.4.1.2 A documented procedure for archiving copies of the compliance artifacts of the supplied software - where the archive is planned to exist for a reasonable period of time since the last offer of the supplied software; or as required by the identified licenses (whichever is longer). Records exist that demonstrate the procedure has been properly followed.~~

~~**Rationale:**~~

~~To ensure reasonable commercial efforts have been instituted in the preparation of the compliance artifacts that accompany the supplied software, as required by the identified licenses.~~

3.6 Adherence to the guideline requirements

3.6.1 Assurance

In order for a program to be deemed conformant with this reference guide, the organization shall affirm that the program satisfies the requirements presented in this document.

Verification material(s):

- 3.6.1.1 A document affirming the program specified in §3.2.4 satisfies all the requirements of this document.

Rationale:

To ensure that if an organization declares that it has a program that is OpenChain conforming, that such program has met all the requirements of this document. The mere meeting of a subset of these requirements is not considered sufficient.

3.6.2 Duration

A program that is conformant with this version of the reference guide shall last 18 months from the date conformance validation was obtained.

Verification material(s):

- 3.6.2.1 A document affirming the program meets all the requirements of this document, within the past 18 months of obtaining conformance validation.

Rationale:

It is important for a program to remain current with the reference guide requirements if an organization wants to assert conformance over time. This requirement ensures that the program's supporting processes and controls do not erode if an organization continues to assert program conformance over time.