OPENCHAIN

OpenChain Security Assurance Reference Guide

Version 1.0

Establishing trust in the Open Source from which Software Solutions are built

Contents

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 for Open Source Compliance, focuses on establishing trust around Open Source license compliance. As a natural next step in support of the broader mission of establishing trust in Open Source, the working group developed this guide to identify and describe the minimum core set of requirements that every Security Assurance program should satisfy with respect to the use of Open Source software. Initially the scope is limited to ensuring that an organization vets the Open Source used with regard to known publicly available security vulnerability issues (e.g., CVEs, github dependency alerts, package manager alerts and so forth). The guide's scope may be expanded overtime based on community feedback.

Conformance with this reference guide provides assurance that an organization has a Program in place that takes the expected steps necessary to establish a trusted 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 industries to choose specific policy and process content that fits their size, goals and Program scope. For instance, a conformant Program may address a single product line or the entire organization.

This introduction describes the guide's purpose. Section 2 defines key terms used throughout this document. Section 3 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).

OpenChain Security Assurance Reference Guide

1 Scope

This document specifies the key requirements of a quality Open Source Security Assurance Program that establishes 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.

2.1 CVE

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. The CVE database is sponsored by the US Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA).

2.2 Known Vulnerability (Vulnerabilities)

Security vulnerabilities that were discovered in Open Source components that are publicly available. That would include any publicly published vulnerabilities including but not limited to CVEs, github dependency alerts, package manager alerts and so forth.

2.3 Open Source

software subject to one or more licenses that meet the Open Source Definition published by the Open Source Initiative (see opensource.org/osd) or the Free Software Definition published by the Free Software Foundation (see gnu.org/philosophy/free-sw.html) or similar license

2.4 Program

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

2.5 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.

2.6 Security Assurance

the confidence that a system meets the requirements for security best practices and is resilient against Known Vulnerabilities.

2.7 SPDX

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

2.8 Supplied Software

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

2.9 Verification Materials

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

3 Requirements

3.1 Program foundation

3.1.1 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 Security 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 Security Assurance policy. Although no requirements are provided here on what should be included in the policy, other sections may impose additional requirements.

3.1.2 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 Program Participants.
- ☐ 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.

3.1.3 Awareness

The organization shall ensure that the Program Participants are aware of:

- The Open Source Security Assurance policy;
- Relevant Program 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.

3.1.4 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.

3.1.5 Standard Practice Implementation

- Organization knowledge of Known Vulnerabilities exists
- 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

A process shall exist for the Security Assurance methods listed above.

Verification material(s):

- 3.1.5.1 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.

3.2 Relevant tasks defined and supported

3.2.1 Access

Maintain a process to effectively respond to Known Vulnerability external inquiries. Publicly identify a means by which a third party can inquire about a Known Vulnerability 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 Known Vulnerability inquiry (e.g., via a published contact email address – security@company.com, opensource@company.com, …).
- ☐ 3.2.1.2 An internal documented procedure for responding to third party Known Vulnerability inquiries exists

Rationale:

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

3.2.2 Effectively resourced

Identify and Resource Program Task(s):

- Assign accountability to ensure the successful execution of Program tasks.
- Program tasks are sufficiently resourced:
  - Sufficient 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 Known 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 expertise available to address identified Known Vulnerabilities.
- ☐ 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 Known Vulnerability 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.

3.3 Open Source content review and approval

3.3.1 Bill of materials

A process shall exist for creating and maintaining a bill of materials that includes each Open Source component 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 Known Vulnerabilities exist

3.3.2 Security Assurance

- For each Open Source component in the bill of materials for the Supplied Software release under review
  - Apply method for detecting existence of Known Vulnerabilities
  - For each identified Known Vulnerability assign a risk/impact score
  - Depending on the risk/impact score take the appropriate action (e.g., contact customers if warranted, upgrade component, no further action, …)
  - If Known Vulnerability is present in previously distributed Supplied Software, depending on the risk/impact score take the appropriate action (e.g., contact customers if warranted)
- Post software solution release - ability to later identify newly reported Known Vulnerabilities that may impact a Supplied Software solution and to respond accordingly

Verification material(s):

- ☐ 3.3.2.1 A documented procedure for handling detection and resolution of Known Vulnerabilities for the Open Source components of the Supplied Software.
- ☐ 3.3.2.2 For each Open Source 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 from which the Supplied Software is comprised. That a procedure exists to support this activity and that the procedure is followed.

3.4 Adherence to the guideline requirements

3.4.1 Completeness

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):

- ☐ 3.4.1.1 A document affirming the Program specified in §3.1.4 satisfies all the requirements of this document.

Rationale:

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

3.4.2 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):

- 3.4.2.1 A document affirming the Program meets all the requirements of this guide, 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.