Code Protection Plan

Revision

Version 8
4/22/24 3:47 PM

SME

Charles Wilson

Abstract

This document describes a set of practices used to ensure that the code used in conjunction with safety-critical, cyber-physical system development and deployment is protected from damage or loss.

Group / Owner

Devops / Information Systems Security Developer

Motivation

This document is motivated by the need to adopt best practices regarding management of source code to allow for certification of compliance to standards such as **ISO/SAE 21434** and **ISO 26262**.

License

Overview

As the source code needed to create software represents the core of safety-critical, cyber-physical systems and the systems supporting them, it is essential that proper cybersecurity controls be applied throughout the development lifecycle.

From ***SSDF PS.1.1***:

> Store all forms of code, including source code and executable code, based on the principle of least privilege so that only authorized personnel have the necessary forms of access. The protection needed will vary based on the nature of the code. For example, some code may be intended for public access, in which case its integrity and availability should be protected; other code may also need its confidentiality protected.

Practices

Source Code Repository

Store all source code in a code repository and restrict access to it.

Store organization, open-source, and third-party materials in separate repositories.

Use version control features of the repository to track all changes made to code with accountability to the individual developer account.

Code Signing

Use code signing to help protect the integrity and provenance of executables.

Use cryptographic hashes to help protect the integrity of files.

Note: A process for code signing is presented in the **Code Signing** [13] elaboration document.

Software Bill of Materials

Create and maintain a software bill of materials (SBOM) for each piece of software stored in the repository.

Use standard elements (SWID for executables, SPDX for components [source, libraries]) to construct the SBOM.

Use the elements of the SBOM to ensure that the products built are using approved source material.

Note: The SBOM lifecycle is presented in the **Software Bill of Materials Lifecycle** [14] elaboration document.

Source Code Archive

Create and maintain an archive of the source code used to produce a product release.

References

1. **List of Approved Tools** (AVCDL secondary document)
2. BSA: IA.1, IA.2-2, SM.4-1 (**BSA: Framework for Secure Software**)
   https://www.bsa.org/files/reports/bsa_software_security_framework_web_final.pdf
3. IDASOAR [16]: Fact Sheet 25 (**IDA State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation 2016 - Development/Sustainment Version Control**)
   https://www.ida.org/-/media/feature/publications/s/st/stateoftheart-resources-soar-for-software-vulnerability-detection-test-and-evaluation-2016/p-8005.ashx
4. NISTCSF: PR.AC-4 (**Identity Management, Authentication and Access Control: Access permissions**)
   https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
   a. CIS CSC 3, 5, 12, 14, 15, 16, 18
      https://learn.cisecurity.org/cis-controls-download
   b. COBIT 5 DSS05.04
      https://www.isaca.org/resources/cobit
   c. ISA 62443-2-1:2009 4.3.3.7.3 ISA 62443-3-3:2013 SR 2.1
      https://www.isa.org/products/isa-62443-2-1-2009-security-for-industrial-automat
   d. ISO/IEC 27001:2013 A.6.1.2, A.9.1.2, A.9.2.3, A.9.4.1, A.9.4.4, A.9.4.5
      https://www.iso.org/standard/54534.html
   e. NIST SP 800-53 Rev. 5 AC-1, AC-2, AC-3, AC- 5, AC-6, AC-14, AC-16, AC-24
      https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
5. PCISSLRAP: 6.1 (**Payment Card Industry (PCI) Security Standards Council, Secure Software Lifecycle (Secure SLC) Requirements and Assessment Procedures**)
   https://www.pcisecuritystandards.org/documents/Secure-Software-Life-Cycle-(SLC)-Program-Guide-v1.pdf
6. SCSIC: Vendor Software Delivery Integrity Controls, Vendor Software Development Integrity Controls (**SAFECode Software Integrity Controls**)
   https://safecode.org/publication/SAFECode_Software_Integrity_Controls0610.pdf
7. SP80064: 3.1.3.5 (**NIST SP 800-160 Systems Security Engineering**)
8. **The Case for Software Bill of Materials** [video 37m]
   http://video.sonatype.com/watch/k1q2hYfAussHmetReM3Jbm
9. **Software Package Data Exchange® (SPDX®)**
   https://spdx.dev/wp-content/uploads/sites/41/2017/12/spdxversion2.1.pdf
10. ISO 19770-2:2015 **Information technology - IT asset management - Part 2: Software identification tag**
    https://www.iso.org/standard/65666.html
11. NIST IR 8060 **Guidelines for the Creation of Interoperable Software Identification (SWID) Tags**
    https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8060.pdf

12. NIST SP 800-218 (**Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework [SSDF]**)
    https://csrc.nist.gov/publications/detail/sp/800-218/draft
13. **Code Signing** (AVCDL elaboration document)
14. **Software Bill of Materials Lifecycle** (AVCDL elaboration document)