OpenSSF Security Baseline

All your base are belong to us

OpenSSF

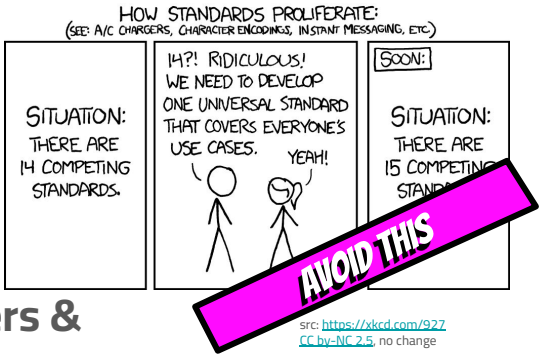
OPEN SOURCE SECURITY FOUNDATION

What we're talking 'bout today!

- OpenSSF Baselines
  - Why we made a new thing
  - What the thing is
  - What the thing includes
  - Why it's awesome / Why you should care
  - Plans for ~~world domination~~ **expansion**
  - How you can help!

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Why are we making yet another thing?



1) Global **cybersecurity legislation is continuing to grow and expand** the compliance obligations for enterprises, manufacturers, and **impacts upstream open source developers & projects**
2) **Existing security frameworks** have two primary **assumptions that are not universally true**: you're already in the security space, or you are on GitHub
3) **Continued lack of trust in open source** despite it being transparent, its **difficult to understand how you can trust it**
    a) transparency does not mean all information is available (or exists)
4) Understanding is key, time and again **security is most successful when it meets developers where they are**, in workflow, in jargon, and in **contextualizing effort with cost-benefit analysis**.

Enter the

OPEN SOURCE PROJECT SECURITY (OSPS) BASELINE

<insert dramatic music>
a collab between the OpenSSF, FINOS, CNCF, LF EU & OpenJS

🗣️ *aa-spuhs bays-line : kindly encouragement and guidance for improving the security of open source projects*

All your Base are belong to us

The OpenSSF Security Baseline was released Feb 2025

Based on a library of well-known cybersecurity frameworks, standards, and global regulations

It includes 40 requirements across 3 levels of maturity covering 8 areas of cyber and application security practices

- Access Control
- Build & Release
- Documentation
- Governance
- Legal
- Quality
- Security Assessment
- Vulnerability Management

https://baseline.openssf.org
https://github.com/ossf/security-baseline

Standards + frameworks-based criteria

ACCESS CONTROL

How is access and changes to the code and pipeline managed?

DOCUMENTATION

What does the project provide for docs, guides, and other artifacts?

LEGAL

What licenses are in place for the project?

GOVERNANCE

What are the project's policies & procedures?

BUILD + RELEASE

How are the CI/CD pipelines configured and managed + build artifact guarantees?

QUALITY

What testing and change management are in place?

SECURITY ASSESSMENT

How does the project evaluate and assess threats and risks?

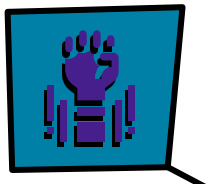
VULN MANAGEMENT

How does the project find, fix, and disclose security vulnerabilities?

Baseline is grounded in international security frameworks

So **“BASELINING”** helps you be compliant!

Global Frameworks

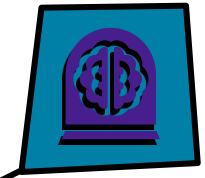


ISO 27001
ISO 27002

International InfoSec frameworks

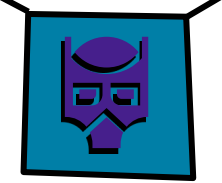
EU CRA + PLD

The Cyber Resilience Act and Product Liability Directive



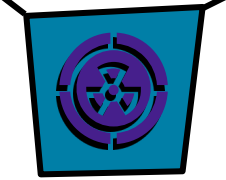
NIST SSDF 1.1

NIST's Secure Software Development Framework



CSA CCM V4

Cloud Security Alliance's Cloud Controls Matrix

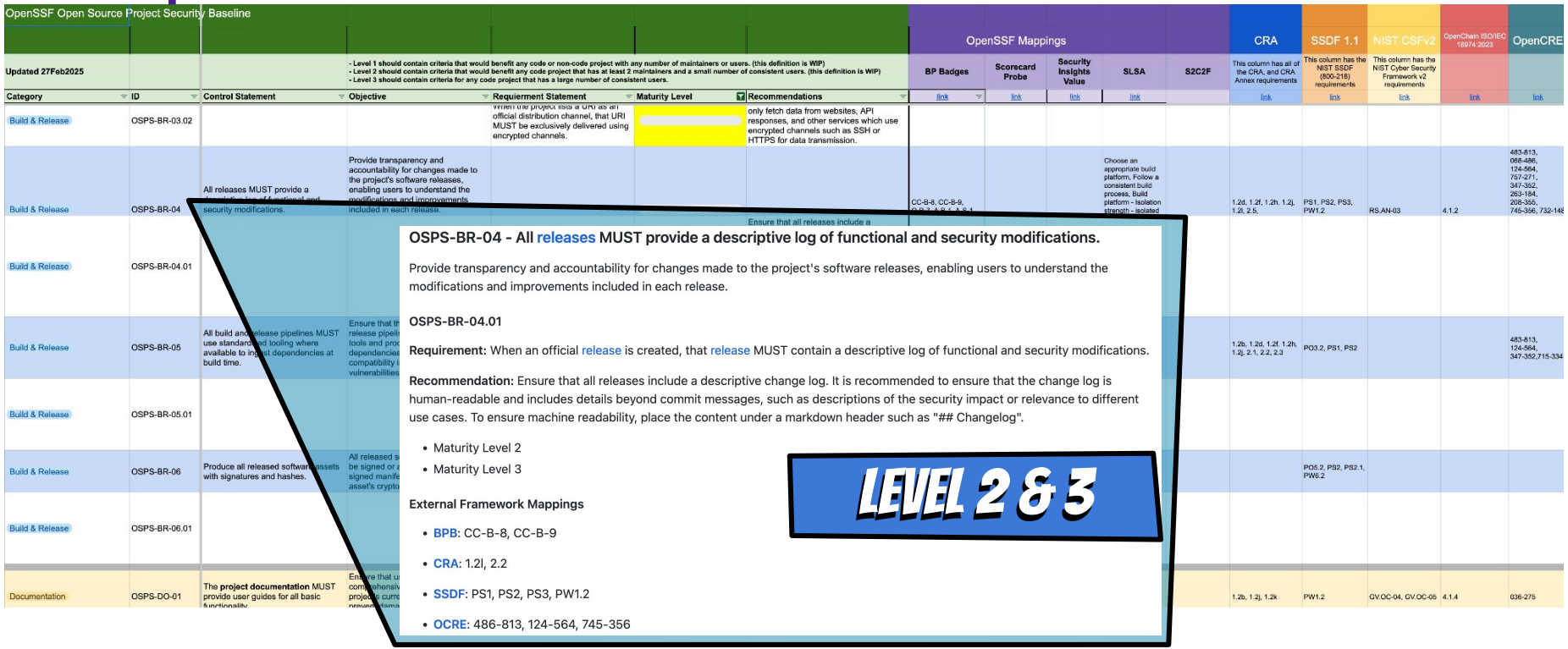


NIST CSF 2.0

NIST's Cybersecurity Framework

...AND MANY OTHERS!

Compliance Crosswalk



OpenSSF Open Source Project Security Baseline

							OpenSSF Mappings					CRA	SSDF 1.1	NIST CSFv2	OpenChain ISO/IEC 18974:2023	OpenCRE
Updated 27Feb2025			- Level 1 should contain criteria that would benefit any code or non-code project with any number of maintainers or users. (this definition is WIP) - Level 2 should contain criteria that would benefit any code project that has at least 2 maintainers and a small number of consistent users. (this definition is WIP) - Level 3 should contain criteria for any code project that has a large number of consistent users.				BP Badges	Scorecard Probe	Security Insights Value	SLSA	S2C2F	This column has all of the CRA, and CRA Annex requirements	This column has the NIST SSDF (800-218) requirements	This column has the NIST Cyber Security Framework v2 requirements		
Category	ID	Control Statement	Objective	Requierment Statement	Maturity Level	Recommendations	link	link	link	link		link	link	link	link	link
Build & Release	OSPS-BR-03.02			When the project lists a URI as an official distribution channel, that URI MUST be exclusively delivered using encrypted channels.		only fetch data from websites, API responses, and other services which use encrypted channels such as SSH or HTTPS for data transmission.										
Build & Release	OSPS-BR-04	All releases MUST provide a descriptive log of functional and security modifications.	Provide transparency and accountability for changes made to the project's software releases, enabling users to understand the modifications and improvements included in each release.			Ensure that all releases include a	CC-B-8, CC-B-9,			Choose an appropriate build platform. Follow a consistent build process. Build platform - Isolation strength - isolated		1.2d, 1.2f, 1.2h, 1.2j, 1.2l, 2.5,	PS1, PS2, PS3, PW1.2	RS.AN-03	4.1.2	483-813, 068-486, 124-564, 757-271, 347-352, 263-184, 208-355, 745-356, 732-148
Build & Release	OSPS-BR-04.01															
Build & Release	OSPS-BR-05	All build and release pipelines MUST use standardized tooling where available to install dependencies at build time.	Ensure that the release pipeline tools and proc dependencies compatibility vulnerabilities									1.2b, 1.2d, 1.2f, 1.2h, 1.2j, 2.1, 2.2, 2.3	PO3.2, PS1, PS2			483-813, 124-564, 347-352,715-334
Build & Release	OSPS-BR-05.01															
Build & Release	OSPS-BR-06	Produce all released software assets with signatures and hashes.	All released s be signed or a signed manife asset's crypto										PO5.2, PS2, PS2.1, PW6.2			
Build & Release	OSPS-BR-06.01															
Documentation	OSPS-DO-01	The **project documentation** MUST provide user guides for all basic	Ensure that u comprehensiv project's curre									1.2b, 1.2j, 1.2k	PW1.2	GV.OC-04, GV.OC-05	4.1.4	036-275

OSPS-BR-04 - All releases MUST provide a descriptive log of functional and security modifications.

Provide transparency and accountability for changes made to the project's software releases, enabling users to understand the modifications and improvements included in each release.

OSPS-BR-04.01

Requirement: When an official release is created, that release MUST contain a descriptive log of functional and security modifications.

Recommendation: Ensure that all releases include a descriptive change log. It is recommended to ensure that the change log is human-readable and includes details beyond commit messages, such as descriptions of the security impact or relevance to different use cases. To ensure machine readability, place the content under a markdown header such as "## Changelog".

- Maturity Level 2
- Maturity Level 3

External Framework Mappings

- BPB: CC-B-8, CC-B-9
- CRA: 1.2l, 2.2
- SSDF: PS1, PS2, PS3, PW1.2
- OCRE: 486-813, 124-564, 745-356

LEVEL 2 & 3

https://docs.google.com/spreadsheets/d/1an5mx3rayoz3JRFUepD56zgprpwXBXBG70fVZvIMCpA/

What are these levels you speak of?

- “Universal security floor” for all open source - great for single maintainer or early maturity projects
  - Are you a Foundation? the level 1 baseline should be your first set of criteria for maturing projects (or even accepting projects).
- Let me get my cli, i got this - good for projects with 2 - 6 maintainers and maturing
- Security *flex* - good for highly mature projects that consider security a core competency
  - Are you in a Foundation with project resources? You should strive for this one.

https://baseline.openssf.org
https://github.com/ossf/security-baseline

Bringing together your favourite OSSF tools!

Best Practices Badges

Interview-based verification of security practices

Scorecard

Automated checking of 19 security aspects of a project

Security Insights

Machine-readable output of assertions

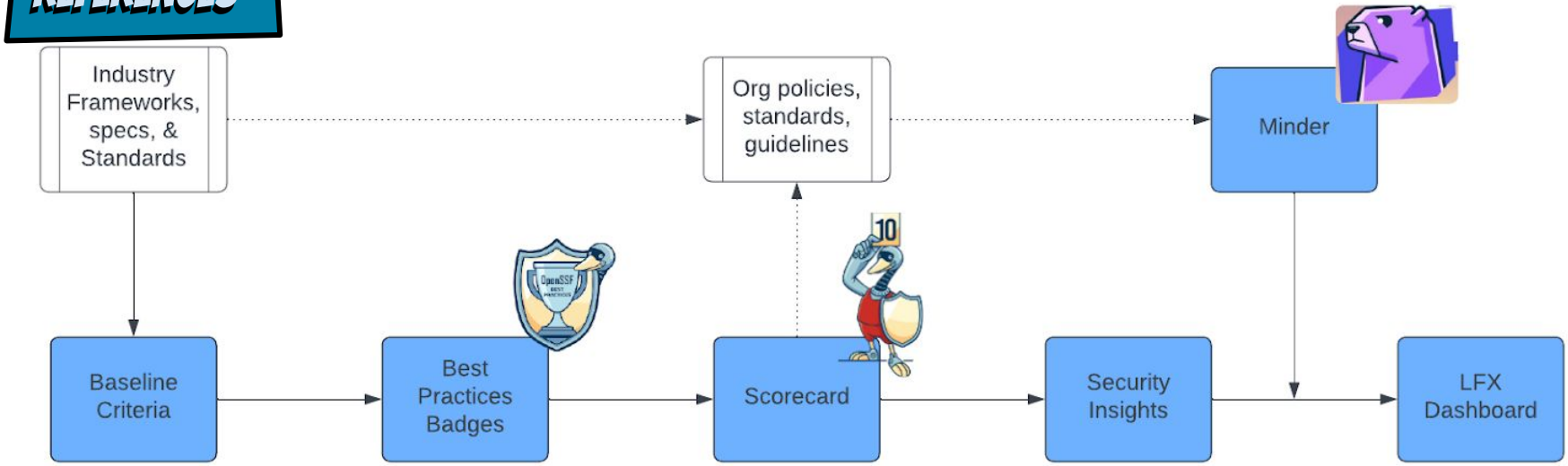
Minder

- Policy-driven decisions on desired security criteria

Future Workflow

REFERENCES

POLICY-BASED DECISIONS



CATALOGUE

ONBOARDING

AUTOMATED EVALUATION

MACHINE-READABLE ATTESTATIONS

COMPLIANCE DASHBOARD

What's Next for the Baseline?

- Augment Recommendations, templates, and guides
- Add Automation to ease adoption of the Baseline Criteria
- Develop Evidence and Attestation collection/publication so downstream can leverage upstream artifacts for their own compliance needs
- Integrate into LFX for "single pane of glass" holistic view
- Define Reference Architecture that shows end-to-end view of Baseline & other components

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

How you can help!

Feedback

Take a look and let us know what isn't clear

Contribute

Submit your suggested changes, contribute automation and tooling

Implement

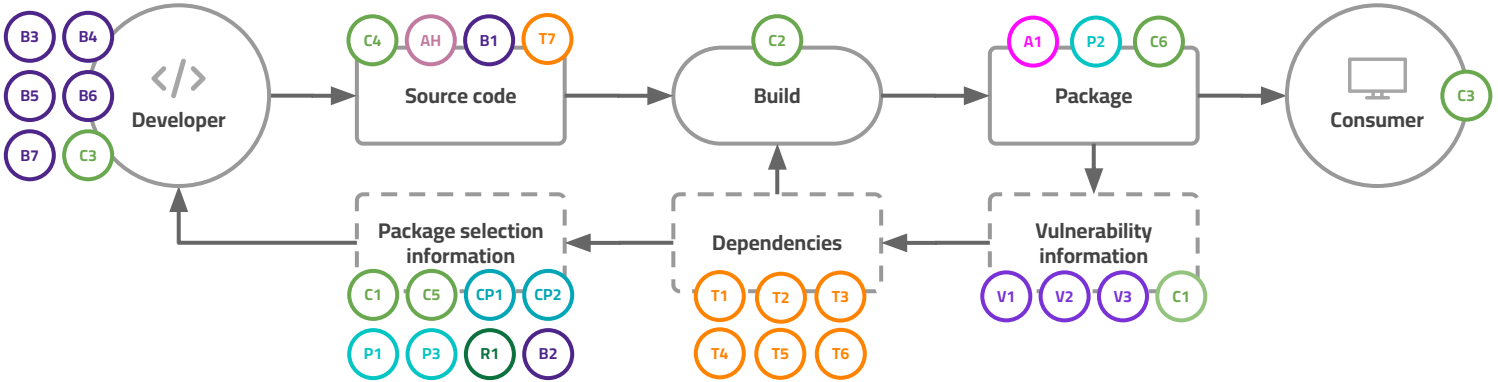
Adopt it for your project or projects in your Foundation

Thank You

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Legal Notice

OpenSSF Technical Initiatives Landscape



Best Practices

B1. OpenSSF Best Practices Badge project
B2. OpenSSF Scorecard project
B3. Education SIG
B4. Memory Safety SIG
B5. C/C++ Compiler Options SIG
B6. Python Hardening SIG
B7. Security Baseline SIG

Supply Chain Integrity

C1. Security Insights project
C2. SLSA project
C3. S2C2F project
C4. Gittuf project
C5. GUAC project
C6. Zarf project M

Security Tooling

T1. SBOM Everywhere SIG
T2. OSS Fuzzing SIG
T3. SBOMit project
T4. Protobom project
T5. bomctl project
T6. Fuzz Introspector project
T7. Minder project

Securing Critical Projects

CP1. criticality_score project
CP2. Package Analysis project

Projects

P1. Alpha & Omega project
P2. Sigstore
P3. Core Toolchain Infrastructure (CTI)

DevRel Community

Diversity, Equity, & Inclusion

Vulnerability Disclosures

V1. CVD Guides SIGs
V2. OSV Schema project
V3. OpenVEX SIG
OpenVEX Project

AI/ML Security

A1. Model Signing SIG

Securing Software Repositories

R1. RSTUF Project

Assembling a compelling security story

Global cybersecurity legislation is continuing to grow and expand the compliance obligations for enterprises, manufacturers, and impacts upstream open source developers & projects

The OpenSSF has 8 working groups, 19 software projects, and multiple guides, practices, and other artifacts

Baseline Criteria by Level

OSPS Level 1

18 requirements that a single-person project should be able to implement

Level 1

OSPS-AC-01.01: When a user attempts to access a sensitive resource in the project's version control system, the system MUST require the user to complete a multi-factor authentication process.

OSPS-AC-02.01: When a new collaborator is added, the version control system MUST require manual permission assignment, or restrict the collaborator permissions to the lowest available privileges by default.

OSPS-AC-03.01: When a direct commit is attempted on the project's primary branch, an enforcement mechanism MUST prevent the change from being applied.

OSPS-AC-03.02: When an attempt is made to delete the project's primary branch, the version control system MUST treat this as a sensitive activity and require explicit confirmation of intent.

OSPS-BR-01.01: When a CI/CD pipeline accepts an input parameter, that parameter MUST be sanitized and validated prior to use in the pipeline.

OSPS-BR-03.01: When the project lists a URI as an official project channel, that URI MUST be exclusively delivered using encrypted channels.

OSPS-DO-01.01: When the project has made a release, the project documentation MUST include user guides for all basic functionality.

OSPS-DO-02.01: When the project has made a release, the project documentation MUST include a guide for reporting defects.

OSPS-GV-02.01: While active, the project MUST have one or more mechanisms for public discussions about proposed changes and usage obstacles.

OSPS-GV-03.01: While active, the project documentation MUST include an explanation of the contribution process.

OSPS-LE-02.01: While active, the license for the source code MUST meet the OSI Open Source Definition or the FSF Free Software Definition.

OSPS-LE-02.02: While active, the license for the released software assets MUST meet the OSI Open Source Definition or the FSF Free Software Definition.

OSPS-LE-03.01: While active, the license for the source code MUST be maintained in the corresponding repository's LICENSE file, COPYING file, or LICENSE/ directory.

OSPS-LE-03.02: While active, the license for the released software assets MUST be included in the released source code, or in a LICENSE file, COPYING file, or LICENSE/ directory alongside the corresponding release assets.

OSPS-QA-01.01: While active, the project's source code repository MUST be publicly readable at a static URL.

OSPS-QA-01.02: The version control system MUST contain a publicly readable record of all changes made, who made the changes, and when the changes were made.

OSPS-QA-02.01: When the package management system supports it, the source code repository MUST contain a dependency list that

OSPS Level 2

24 requirements that should be attainable by 2-6 person projects

Level 2

OSPS-AC-04.01: When a CI/CD task is executed with no permissions specified, the project's version control system MUST default to the lowest available permissions for all activities in the pipeline.

OSPS-BR-02.01: When an official release is created, that release MUST be assigned a unique version identifier.

OSPS-BR-04.01: When an official release is created, that release MUST contain a descriptive log of functional and security modifications.

OSPS-BR-05.01: When a build and release pipeline ingests dependencies, it MUST use standardized tooling where available.

OSPS-BR-06.01: When an official release is created, that release MUST be signed or accounted for in a signed manifest including each asset's cryptographic hashes.

OSPS-DO-06.01: When the project has made a release, the project documentation MUST include a description of how the project selects, obtains, and tracks its dependencies.

OSPS-GV-01.01: While active, the project documentation MUST include a list of project members with access to sensitive resources.

OSPS-GV-01.02: While active, the project documentation MUST include descriptions of the roles and responsibilities for members of the project.

OSPS-GV-03.02: While active, the project documentation MUST include a guide for code contributors that includes requirements for acceptable contributions.

OSPS-LE-01.01: While active, the version control system MUST require all code contributors to assert that they are legally authorized to make the associated contributions on every commit.

OSPS-QA-03.01: When a commit is made to the primary branch, any automated status checks for commits MUST pass or be manually bypassed.

OSPS-QA-06.01: Prior to a commit being accepted, the project's CI/CD pipelines MUST run at least one automated test suite to ensure the changes meet expectations.

OSPS-SA-01.01: When the project has made a release, the project documentation MUST include design documentation demonstrating all actions and actors within the system.

OSPS-SA-02.01: When the project has made a release, the project documentation MUST include descriptions of all external software interfaces of the released software assets.

OSPS-SA-03.01: When the project has made a release, the project MUST perform a security assessment to understand the most likely and impactful potential security problems that could occur within the software.

OSPS-VM-01.01: While active, the project documentation MUST include a policy for coordinated vulnerability reporting, with a clear timeframe for response.

OSPS Level 3

9 requirements for mature, well-resourced projects should be able to achieve

Level 3

OSPS-AC-04.02: When a job is assigned permissions in a CI/CD pipeline, the source code or configuration MUST only assign the minimum privileges necessary for the corresponding activity.

OSPS-BR-02.01: When an official release is created, all assets within that release MUST be clearly associated with the release identifier or another unique identifier for the asset.

OSPS-DO-03.01: When the project has made a release, the project documentation MUST contain instructions to verify the integrity and authenticity of the release assets.

OSPS-DO-04.01: When the project has made a release, the project documentation MUST include a descriptive statement about the scope and duration of support for each release.

OSPS-DO-05.01: When the project has made a release, the project documentation MUST provide a descriptive statement when releases or versions will no longer receive security updates.

OSPS-GV-04.01: While active, the project documentation MUST have a policy that code contributors are reviewed prior to granting escalated permissions to sensitive resources.

OSPS-QA-02.02: When the project has made a release, all compiled released software assets MUST be delivered with a software bill of materials.

OSPS-QA-04.02: When the project has made a release comprising multiple source code repositories, all subprojects MUST enforce security requirements that are as strict or stricter than the primary codebase.

OSPS-QA-06.02: While active, project's documentation MUST clearly document when and how tests are run.

OSPS-QA-06.03: While active, the project's documentation MUST include a policy that all major changes to the software produced by the project should add or update tests of the functionality in an automated test suite.

OSPS-QA-07.01: When a commit is made to the primary branch, the project's version control system MUST require at least one non-author approval of the changes before merging.

OSPS-SA-03.02: When the project has made a release, the project MUST perform a threat modeling and attack surface analysis to understand and protect against attacks on critical code paths, functions, and interactions within the system.

OSPS-VM-04.02: While active, any vulnerabilities in the software components not affecting the project MUST be accounted for in a VEX document, augmenting the vulnerability report with non-exploitability details.

OSPS-VM-05.01: While active, the project documentation MUST include a policy that defines a threshold for remediation of SCA findings related to vulnerabilities and licenses.

OSPS-VM-05.02: While active, the project documentation MUST include a policy to address SCA violations prior to any release.

OSPS-VM-05.03: While active, all changes to the project's codebase MUST be automatically evaluated against a documented policy