CRA Resource Inventory: references relevant to the development and integration of open source under the Cyber Resilience Act

The goal of this document is to provide a comprehensive list of resources that are relevant to the obligations of open source software stewards and manufacturers when it comes to the development and integration of open source under the Cyber Resilience Act (CRA) and to the interactions between all of the different stakeholders. The underlying purpose is to provide specification and standardization efforts with easy access to documented industry and community best practices.

Note: the description of each resource has been generated using a large language model and verified for accuracy. See Annex I - LLM Usage below for the prompts and tools used.

Status of this document

This document is a deliverable of the Cyber Resilience SIG of the Open Regulatory Compliance Working Group (ORC) of the Eclipse Foundation. It was approved to be released as version 1.0 by the Cyber Resilience SIG on May 12, 2025. It represents the consensus of ORC and its members.

This document is developed in the open, on GitHub. To contribute to future revisions of this document or submit errata, please send a pull request or open an issue.

Table of Content

1. Principles of security resilience

This section contains references which are relevant to the requirements expressed in Annex I, Part I(1) of the CRA. It corresponds to the horizontal type "A" standard number 1 of the European Commission's standardisation request to the ESOs:

> *Products with digital elements shall be designed, developed and produced in such a way that they ensure an appropriate level of cybersecurity based on the risks.*

The impact on open source software stewards is limited to a partial obligation to "foster development of secure product" expressed in Article 24(1).

Technical Guideline TR-03183: Cyber Resilience Requirements for Manufacturers and Products - Part 1: General requirements

Metadata

URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03183/BSI-TR-03183-1-0_9_0.pdf?__blob=publicationFile&v=4

Publisher: Federal Office for Information Security (BSI), Germany

License: Not specified

Type: Technical Guideline (draft regulatory standard)

Publication date: 2024

The Technical Guideline TR-03183 from Germany's BSI delineates a set of cyber resilience requirements for manufacturers and product, aligned with the EU's

upcoming Cyber Resilience Act (CRA). The guideline translates the CRA's regulatory expectations into concrete technical criteria and practices. By providing this framework in advance of the CRA's full enforcement, the document helps industry stakeholders bolster software security and compliance, ensuring that products with digital elements are developed and maintained in line with emerging European cybersecurity standards. Part 1 focuses on the general requirements of the CRA (e.g. applying a risk-based approach and securing the software development lifecycle).

ETSI 303 645 - Cyber Security for Consumer Internet of Things: Baseline Requirements

Metadata

URL: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf

Publisher: ETSI

License: ?

Type: formal

Publication date: 2021

ETSI EN 303 645 is a European standard specifying baseline cybersecurity requirements for consumer Internet of Things devices. It enumerates a set of fundamental practices that manufacturers of smart home and wearable devices, for instance, should implement to significantly improve security. Notable provisions include: avoiding universal default passwords (each device must have a unique or user-set credential), providing a means to manage vulnerability disclosure (publish a contact for security issues), keeping software updated (and informing consumers of update duration), securely storing sensitive data, and minimizing exposed attack surfaces. The standard, first released in 2020 (and updated in v2.1.1 to refine requirements), has 13 key provisions in total. While voluntary, EN 303 645 has

become highly influential; its guidelines have been adopted or referenced by governments and industry schemes globally (such as in the UK's and Australia's IoT codes of practice and Singapore's CLS levels). In summary, ETSI EN 303 645 serves as a baseline "security hygiene" checklist for IoT products to reduce common vulnerabilities and make consumer devices more resilient against attacks like botnet recruitment or data breaches.

ISO 31000:2018 Risk management — Guidelines

Metadata

URL: https://www.iso.org/standard/65694.html

Publisher: ISO

License: ?

Type: Formal

Publication date: 2018

ISO 31000:2018 is an international standard that provides principles and generic guidelines for effective risk management in organizations. Rather than prescribing a specific technique, it outlines a framework and process that organizations can use to identify, assess, and treat risks systematically. Key concepts include establishing the context of risk (understanding the internal and external environment), performing structured risk assessments (identifying risks, analyzing their likelihood and impact, evaluating priorities), and then treating the risks with appropriate controls or mitigation strategies. The standard emphasizes that risk management should be iterative and embedded into organizational processes, decision-making, and culture. By following ISO 31000's guidelines, organizations of any type can improve their ability to manage uncertainty, minimize losses, and maximize opportunities in a rational, auditable way.

NIST 800-37: Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for

Security and Privacy

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/37/r2/final

Publisher: NIST

License: ?

Type: informal

Publication date: 2018

NIST Special Publication 800-37 Revision 2 is the authoritative guide on applying a Risk Management Framework (RMF) to information systems. The RMF provides a disciplined, structured process that integrates security and privacy into the system development life cycle. The steps of the RMF include: Prepare (organizational and system level preparation for risk management), Categorize (determining the system's impact level in terms of confidentiality, integrity, availability), Select (choosing an initial set of security and privacy controls from standards like NIST 800-53 based on that categorization and tailoring them as needed), Implement (putting the controls in place), Assess (evaluating how effectively the controls are implemented), Authorize (a senior official formally accepting the residual risk before the system operates), and Monitor (continuously tracking the system's security posture and control effectiveness over time). Rev. 2 of SP 800-37, published in 2018, updated the process to better support organization-wide risk management, emphasized the importance of privacy, and aligned with the NIST Cybersecurity Framework. It promotes ongoing authorization and continuous monitoring rather than one-time snapshots. Utilizing RMF helps organizations (especially U.S. federal agencies and contractors) ensure that security is not an afterthought but a continuous, lifecycle concern tied to risk decisions.

NIST SP 800-160 Vol. 2 Rev. 1: Developing Cyber-Resilient Systems: A Systems Security Engineering Approach

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/160/v2/r1/final

Publisher: NIST

License: ?

Type: informal

Publication date: 2021

This NIST publication focuses on engineering principles for cyber resilience, complementing traditional cybersecurity (which often emphasizes protection and detection) with strategies to ensure essential functions can withstand and recover from attacks. It adopts a systems security engineering perspective, meaning it embeds resilience into the system design lifecycle. The document introduces a catalog of cyber resiliency techniques and approaches — for example, techniques like redundancy (having backup components), diversity (using different implementations to avoid common-mode failures), segmentation (limiting the blast radius of compromises), deception (confusing or slowing attackers), and graceful degradation (maintaining partial functionality under stress). Rev. 1 updates these concepts and provides use cases and mappings to known frameworks (like mapping resiliency techniques to NIST CSF functions). By applying the guidance in SP 800-160 Vol. 2, system architects and engineers can design systems that continue to operate even while under attack or after compromise, thereby protecting critical missions. The document essentially shifts the focus from just preventing breaches to assuming that incidents will happen and planning how the system will cope and adapt when they do.

NIST SP 800-218: Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/218/final

Publisher: NIST

License: ?

Type: informal

Publication date: 2022

NIST's Secure Software Development Framework is a set of recommended practices for integrating security into the software development life cycle. Version 1.1 of the SSDF (published in 2022) organizes these practices into four groups: Prepare the Organization (ensure the development team has the resources, training, and governance to produce secure code), Protect the Software (establish secure environments and tools, and protect code integrity – e.g., via version control protections and dependency management), Produce Well-Secured Software (implement security design, code, and testing practices – like threat modeling, static analysis, and code review – and address findings), and Respond to Vulnerabilities (define processes to handle internally and externally reported bugs in released software, including patching and communication). Each practice is described at a high level with mappings to more specific standards (like OWASP, ISO, etc.). The SSDF is meant to be adaptable: organizations of any size can implement its core tenets and can integrate them into existing development workflows (waterfall, Agile, DevOps, etc.). By following the SSDF, software producers minimize the introduction of vulnerabilities and establish a workflow to catch and fix security issues early, which ultimately leads to more secure products for end users.

1.1 Risk analysis

EN18031 - Common security requirements for radio equipment

Metadata

URL: https://www.nen.nl/nen-en-18031-1-2024-en-328074

Publisher: NEN-EN

License: ?

Type: formal

Publication date: 2024

EN 18031-1:2024 is a new European harmonized standard developed to support the security provisions of the EU Radio Equipment Directive (RED). As of August 2024, certain radio-connected devices (including many IoT products) must comply with Article 3(3)(d)-(f) of the RED, which mandates network protection, privacy of data, and fraud prevention. EN 18031-1 provides the “common security requirements” that manufacturers can implement to meet these legal obligations. It likely covers a broad set of controls such as requiring authentication for critical functions, ensuring secure data transmission, protecting personal data handled by the equipment, and maintaining software update mechanisms. This standard builds on prior work (like ETSI EN 303 645) but formalizes it under CEN/CENELEC for regulatory conformity. By following EN 18031-1, manufacturers can self-declare or certify that their wireless and IoT products adhere to the necessary cybersecurity baseline, thereby fulfilling the RED requirements and allowing them to be placed on the EU market. Essentially, EN 18031-1 operationalizes the Cyber Resilience Act’s spirit early for radio devices, harmonizing security across all consumer smart products in Europe.

OWASP Threat Modeling Cheat Sheet

Metadata

URL: https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

Publisher: OWASP

License: CC BY-SA 4.0

Type: informal

Publication date: ?

The OWASP Threat Modeling Cheat Sheet is a distilled guide to performing threat modeling, which is the process of systematically identifying and addressing potential threats to an application's security during the design phase. The cheat sheet outlines the essential steps in threat modeling: defining the security objectives, creating an application diagram or understanding the architecture, identifying threats (often using frameworks like STRIDE: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege), and devising mitigations for those threats. It provides tips and best practices, such as focusing on high-risk assets, involving diverse stakeholders (developers, architects, security engineers), and iterating the threat model as the design evolves. By following this concise guide, developers and teams can ensure they consider security early in the development lifecycle—modeling what could go wrong and building in defenses—rather than reacting to issues late in the game.

1.2 Secure design and secure coding principles

OpenStack Secure development guidelines

Metadata

URL: https://security.openstack.org/#secure-development-guidelines
Publisher: OpenStack Community
License: CC-BY-3.0
Type: informal
Publication date: 2015-02-18 – present

The OpenStack Security Team has created a set of secure development guidelines and best practices to help developers avoid common mistakes that could introduce vulnerabilities in the OpenStack platform. These guidelines cover various areas of secure coding (e.g. proper file permissions, input validation, avoiding insecure libraries, using encryption correctly) and serve as a baseline for developers to follow so that OpenStack components are built with security in mind. By adhering to these recommendations, contributors to OpenStack can systematically reduce security weaknesses and improve the overall resilience of the cloud software.

OWASP Cheat Sheet Series

Metadata

URL: https://cheatsheetseries.owasp.org/
Publisher: OWASP
License: CC BY-SA 4.0
Type: informal
Publication date: 2014 – present

The OWASP Cheat Sheet Series is a collection of concise, high-value guides on a wide range of application security topics. Each cheat sheet focuses on a specific area (for example, authentication, injection prevention, error handling, etc.) and distills essential best practices and recommendations in an easy-to-reference

format. Created and maintained by security experts in the OWASP community, the cheat sheets provide developers and defenders with clear guidelines to implement security controls correctly. By following the Cheat Sheet Series, practitioners can quickly learn the do's and don'ts of application security for common scenarios, thereby improving the security of software without needing to wade through extensive documentation.

1.3 Security processes and governance

Compiler Hardening Guide/ C C++ compiler options

Metadata

URL: https://best.openssf.org/Compiler-Hardening-Guides/Compiler-Options-Hardening-Guide-for-C-and-C++.html

Publisher: OpenSSF

License: ?

Type: informal

Publication date: 2024

This guide (hosted by the OpenSSF Best Practices community) outlines how to leverage modern compiler features and settings to produce more secure software binaries in C and C++. It provides practical recommendations on compiler options that enable various hardening mechanisms. Examples include turning on stack canaries and stack overflow protections (-fstack-protector-strong), using address space layout randomization friendly flags (-fPIE for position-independent executables, and linking with -pie), enabling control-flow integrity or forward-edge protections (if supported by the compiler toolchain), enabling warnings and treating them as errors for risky patterns, and using memory sanitization tools in testing builds. The guide may also discuss choices like enabling Fortify Source (to catch

certain buffer overflows at runtime) and using latest standards (e.g. C11/C17 features that improve safety). Each suggested option is explained in terms of what security benefit it provides and any trade-offs (such as performance impact or compatibility). Following this hardening guide helps developers raise the baseline security of their C/C++ applications, making them more resistant to common memory corruption vulnerabilities by catching issues early or making exploitation much more difficult.

NIST Cybersecurity Framework

Metadata

URL: https://www.nist.gov/cyberframework

Publisher: NIST

License: ?

Type: informal

Publication date: Ongoing

The NIST CSF is a high-level framework that helps organizations manage and reduce cybersecurity risk. It is composed of five Core Functions: Identify (know your assets, systems, data, and risks), Protect (implement safeguards like access controls, training, maintenance), Detect (deploy activities to promptly detect anomalies and incidents), Respond (have plans and actions for containing and minimizing incidents), and Recover (restore any capabilities or services impaired by incidents and incorporate lessons learned). Under each function are Categories and Subcategories that outline specific outcomes (for example, under Protect -> Access Control, an outcome is “identities and credentials are managed for authorized devices and users”). The CSF also includes Implementation Tiers to gauge the maturity of risk management practices and a Profile concept to tailor the framework to the organization’s goals and sector. Initially developed for critical infrastructure, the CSF has been widely adopted across industries globally because of its flexibility and clear structure. It’s often used as a starting point for developing

a cybersecurity program or as a communication tool between technical teams and management, since it distills complex security practices into a straightforward, business-aligned format. (Note: A CSF 2.0 update is underway to incorporate more guidance on governance and supply chain risk.)

OWASP DevSecOps Maturity Model (DSOMM)

Metadata

URL: https://dsomm.owasp.org

Publisher: OWASP

License: GPL-3

Type: Maturity Model

Publication date: 2017 – present

The OWASP DevSecOps Maturity Model (DSOMM) is a framework that guides organizations in assessing and enhancing the integration of security within their DevOps processes. It defines multiple domains of software development and operations (such as source code, build pipelines, and deployment environments) and outlines progressive maturity levels for each domain, providing clear benchmarks for improvement. By using DSOMM, teams can identify gaps in their current DevSecOps practices and prioritize security measures—ranging from basic safeguards in continuous integration to advanced, automated security testing—appropriate to their maturity level. This model serves as a reference for building more secure development pipelines, aligning with broader software security best practices and supporting compliance efforts by ensuring that security controls are systematically embedded in the software lifecycle.

OWASP Software Assurance Maturity Model (SAMM)

Metadata

URL: https://owaspsamm.org/

Publisher: OWASP

License: CC-BY-SA 4.0

Type: informal

Publication date: 2009 – present

OWASP SAMM is an open framework that organizations use to evaluate and improve their software security practices. It defines 15 security practices grouped into 5 business functions (such as Governance, Design, Implementation, Verification, and Operations), each with maturity levels and activities. By assessing their maturity against SAMM's criteria, organizations can identify gaps in how they design, develop, and deploy software, and then follow SAMM's guidance to incrementally implement more advanced security practices. The model is technology-agnostic and is updated by the OWASP community to reflect current best practices, making it a versatile roadmap for building a robust software security assurance program.

2. Generic Security Requirements

This section contains references which are relevant to the requirements expressed in Annex I, Part I(2) of the CRA. It corresponds to the horizontal type "B" standards number 2 to 14 of the European Commission's standardisation request to the ESOs.

On the basis of the cybersecurity risk assessment referred to in Article 13(2) and where applicable, products with digital elements shall:

(a) be made available on the market without known exploitable vulnerabilities;
(b) be made available on the market with a secure by default configuration, unless otherwise agreed between manufacturer and business user in relation to a tailor-made product with digital elements, including the possibility to reset the product to its original state;
(c) ensure that vulnerabilities can be addressed through security updates, including, where applicable, through automatic security updates that are installed within an appropriate timeframe enabled as a default setting, with a clear and easy-to-use opt-out mechanism, through the notification of available updates to users, and the option to temporarily postpone them;
(d) ensure protection from unauthorised access by appropriate control mechanisms, including but not limited to authentication, identity or access management systems, and report on possible unauthorised access;
(e) protect the confidentiality of stored, transmitted or otherwise processed data, personal or other, such as by encrypting relevant data at rest or in transit by state of the art mechanisms, and by using other technical means;
(f) protect the integrity of stored, transmitted or otherwise processed data, personal or other, commands, programs and configuration against any manipulation or modification not authorised by the user, and report on corruptions;
(g) process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital

elements (data minimisation);
(h) protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks;
(i) minimise the negative impact by the products themselves or connected devices on the availability of services provided by other devices or networks;
(j) be designed, developed and produced to limit attack surfaces, including external interfaces;
(k) be designed, developed and produced to reduce the impact of an incident using appropriate exploitation mitigation mechanisms and techniques;
(l) provide security related information by recording and monitoring relevant internal activity, including the access to or modification of data, services or functions, with an opt-out mechanism for the user;
(m) provide the possibility for users to securely and easily remove on a permanent basis all data and settings and, where such data can be transferred to other products or systems, ensure that this is done in a secure manner.

These requirements do not apply open source software stewards. However, per Article 25 of the CRA, stewards (as well as developers and users of open source software and other third parties) may participate in voluntary security attestation programmes that assess the conformity of their software to some or all of these requirements.

Hardware Secure Boot

Metadata

URL: https://www.opencompute.org/documents/secure-boot-2-pdf

Publisher: OpenCompute

License:

Type: informal

Publication date:

This guide from the Open Compute Project (OCP) provides design requirements and recommendations for implementing secure boot in hardware systems (like servers or network devices). Secure boot is a mechanism where the system's boot firmware (BIOS/UEFI or similar) will only execute code that is cryptographically signed by a trusted authority. The OCP guide likely details how to establish a root of trust in hardware (such as using a TPM or dedicated secure element to store cryptographic keys), how to sign bootloaders and OS kernels, and how the verification process should work at each stage of the boot chain. It may also discuss managing keys (for example, allowing owners to enroll their own keys or update keys securely) and handling firmware updates in a secure manner (ensuring updates are signed and verified). By adhering to this guide, manufacturers can ensure their devices are protected against low-level malware: even if an attacker has physical access or can alter the boot device, the system will refuse to run untrusted boot code, thus preventing persistent malware like rootkits from taking hold. In summary, the OCP Secure Boot Guide is a blueprint for building devices that only run authentic, untampered software from power-on through full system startup.

NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

Publisher: NIST

License: ?

Type: informal

Publication date: 2020

NIST SP 800-53 is a comprehensive catalog of controls (safeguards and countermeasures) that organizations can apply to protect information systems and personal data. Revision 5, released in 2020, expanded the catalog to integrate privacy fully alongside security and made the language more outcome-based (less federal-centric), so it's usable by a wider audience (government, industry, international). The catalog is organized into families like Access Control, Incident Response, Cryptography, Personnel Security, etc., each containing specific controls. For example, in Access Control, one control is implementing role-based access with the principle of least privilege. Rev. 5 introduced new control families for areas like Supply Chain Risk Management, refined controls for advanced technologies (IoT, mobile), and removed the concept of "low/medium/high baselines" from the document itself (that moved to separate guidance) to focus on the controls. An organization using SP 800-53 will select a subset of these controls based on its risk assessment (often guided by frameworks like SP 800-37 RMF) and then implement and document them. SP 800-53 is widely used not only by U.S. federal agencies (it's the backbone of FedRAMP, DoD, etc. requirements) but also by others as a rich reference of security best practices to draw from when securing systems.

OWASP Application Security Verification Standard

Metadata

URL: https://owasp.org/www-project-application-security-verification-standard/

Publisher: OWASP

License: CC BY-SA 3.0

Type:

Publication date:

The OWASP ASVS is a framework that provides a checklist of application security requirements and controls in a structured manner, serving as a basis for testing web application security. It defines three levels of verification depth: Level 1 (basic, for all applications) through Level 3 (advanced, for the most critical applications). Each level comprises requirements across categories such as Authentication, Access Control, Data Protection, Error Handling, and others. For example, at Level 1, there might be a requirement that the application has no default passwords and uses HTTPS; at higher levels, requirements become more stringent, like implementing strong multi-factor authentication, or using cryptographic modules with specific certifications. Developers and architects can use ASVS during development to ensure they build in these controls, and security testers can use it as a guide for what to verify (it's often used to structure penetration testing or security code review efforts). By using ASVS, organizations get a common language for what it means for an application to be "secure" at a given level, and they can assert compliance to that level. Ultimately, ASVS helps raise the security baseline by providing clear, measurable security criteria for applications.

3. Vulnerability Management

This section contains references which are relevant to:

The requirements expressed in Annex I, Part II of the CRA. This corresponds to the horizontal type "B" standard number 15 of the European Commission's standardisation request to the ESOs:

> *Manufacturers of products with digital elements shall:*
> *(1) identify and document vulnerabilities and components contained in products with digital elements, including by drawing up a software bill of materials in a commonly used and machine-readable format covering at the very least the top-level dependencies of the products;*
> *(2) in relation to the risks posed to products with digital elements, address and remediate vulnerabilities without delay, including by providing security updates; where technically feasible, new security updates shall be provided separately from functionality updates;*
> *(3) apply effective and regular tests and reviews of the security of the product with digital elements;*
> *(4) once a security update has been made available, share and publicly disclose information about fixed vulnerabilities, including a description of the vulnerabilities, information allowing users to identify the product with digital elements affected, the impacts of the vulnerabilities, their severity and clear and accessible information helping users to remediate the vulnerabilities; in duly justified cases, where manufacturers consider the security risks of publication to outweigh the security benefits, they may delay making public information regarding a fixed vulnerability until after users have been given the possibility to apply the relevant patch;*
> *(5) put in place and enforce a policy on coordinated vulnerability disclosure;*

(6) take measures to facilitate the sharing of information about potential vulnerabilities in their product with digital elements as well as in third-party components contained in that product, including by providing a contact address for the reporting of the vulnerabilities discovered in the product with digital elements;

(7) provide for mechanisms to securely distribute updates for products with digital elements to ensure that vulnerabilities are fixed or mitigated in a timely manner and, where applicable for security updates, in an automatic manner;

(8) ensure that, where security updates are available to address identified security issues, they are disseminated without delay and, unless otherwise agreed between a manufacturer and a business user in relation to a tailor-made product with digital elements, free of charge, accompanied by advisory messages providing users with the relevant information, including on potential action to be taken.

The requirement for open source software stewards to "put in place and document a cybersecurity policy" expressed in Article 24(1) of the CRA. It has no corresponding standard request in the European Commission's standardisation request to the ESOs.

1. Open-source software stewards shall put in place and document in a verifiable manner a cybersecurity policy to foster the development of a secure product with digital elements as well as an effective handling of vulnerabilities by the developers of that product. That policy shall also foster the voluntary reporting of vulnerabilities as laid down in Article 15 by the developers of that product and take into account the specific nature of the open-source software steward and the legal and organisational arrangements to which it is subject. That policy shall, in particular, include aspects related to documenting, addressing and remediating vulnerabilities

and promote the sharing of information concerning discovered vulnerabilities within the open-source community.

3.1 Vulnerability management specifications and policy templates

OpenSSF Security Policy Templates

Metadata

URL: https://github.com/ossf/oss-vulnerability-guide/tree/main/templates/security_policies

Publisher: OpenSSF

License: Apache-2.0

Type: Policy Template

Publication date: 2022- present

This resource is a collection of security policy templates provided by the Open Source Security Foundation's Open Source Software Vulnerability Guide project to help open source maintainers create robust security policies for their projects. The templates include ready-made text and structure for documents like a project's security policy or "SECURITY.md," vulnerability reporting guidelines, and coordinated disclosure procedures, all of which maintainers can adapt to their specific needs. By offering standardized language covering aspects such as how to report vulnerabilities, expected response times, and how fixes and advisories will be communicated, these templates lower the barrier for projects to implement best practices in vulnerability management and communication. They are directly relevant to improving open source software resilience and assist projects in meeting expectations set by users and regulations (for instance, the EU Cyber Resilience Act's call for clear vulnerability disclosure policies), ensuring that even

smaller projects can align with industry standards in security process documentation.

OpenSSF Outbound Vulnerability Disclosure Policy Template

Metadata

URL: https://github.com/ossf/wg-vulnerability-disclosures/blob/main/docs/Outbound_Vulnerability_Disclosure_Policy_template.md

Publisher: OpenSSF

License: Apache-2.0

Type: Policy Template

Publication date: 2024- present

The Open Source Security Foundation (OpenSSF) Outbound Vulnerability Disclosure Policy Template is a model policy document designed to guide organizations in responsibly disclosing security vulnerabilities they discover in external projects or products. This template adheres to the “Model Outbound Vulnerability Disclosure Policy” (version 0.1.1) and clarifies processes distinct from inbound vulnerability handling, focusing on how an organization should report a discovered vulnerability to the affected third party or vendor. It outlines recommended steps such as initial notification, coordination with the affected product’s security team or a relevant CERT, setting timelines for remediation and public disclosure, and conditions for involving outside stakeholders (for instance, regulators or the broader security community) if a vulnerability remains unaddressed. By following this template, organizations can establish a clear and consistent outbound disclosure practice, which contributes to the overall health of the software ecosystem and aligns with broader cybersecurity norms and compliance considerations that encourage transparency in vulnerability management across supply chains.

NIST SP 800-231: Bug Framework (BF): Formalizing Cybersecurity Weaknesses and Vulnerabilities

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/231/final

Publisher: NIST

License: ?

Type: informal

Publication date: 2024

NIST's Bug Framework (BF) is an effort to formalize how we describe software weaknesses and vulnerabilities. Unlike vulnerability databases (which catalog specific instances of bugs in products), the BF provides a methodology and language to define classes of bugs in a rigorous way. It builds on concepts from existing taxonomies like CWE (Common Weakness Enumeration) but adds formal structure, making it easier to see the relations between bugs. For example, the framework may define a hierarchy or mapping of how a higher-level weakness (say "Improper Input Validation") can manifest as different lower-level bug types in certain contexts. It likely introduces a model to reason from root causes to impacts. The intent is to aid tools and researchers: with a formal bug definition language, static analysis tools or formal methods can more precisely identify and categorize issues, and data analysts can better aggregate and compare vulnerability information across different sources. In essence, SP 800-231 is bringing scientific rigor to how we classify software flaws, which can improve communication (everyone uses terms the same way), allow for more automation in vulnerability management, and potentially help discover logical gaps in existing security coverage.

RFC 9116

Metadata

URL: https://www.rfc-editor.org/rfc/rfc9116

Publisher: IETF

License: IETF Trust Legal Provisions (TLP)

Type: RFC

Publication date: 2022

RFC 9116 defines the “security.txt” standard, providing a standardized file format for organizations to publish their vulnerability disclosure policies and security contact information. The RFC addresses the absence of clear reporting mechanisms, presenting a machine-readable structure with fields for contact information, encryption details, acknowledgments, policy links, and expiration dates. Widely adopted despite its informational status, RFC 9116 facilitates improved communication between vulnerability reporters and software maintainers, enhancing coordinated disclosure processes and aligning with regulatory requirements such as the EU Cyber Resilience Act.

3.2 Existing open source foundation policies

Apache Softwate Foundation (ASF)

ASF Classification of vulnerabilties

Metadata

URL: https://security.apache.org/blog/severityrating/

Publisher: ASF

License: ASLv2

Type: practice

Publication date: 2023

The Apache Software Foundation (ASF) introduced a standardized vulnerability severity rating system classifying issues into Low, Moderate, Important, and Critical levels based on their exploitability and potential impact. Critical vulnerabilities involve remote exploitation without prerequisites, while lower levels require specific conditions or have less severe implications. This simplified severity rating complements the CVSS system, offering clear, consistent assessments across ASF projects, thereby aiding administrators in prioritizing vulnerability responses and improving coordinated disclosure clarity.

Generic ASF handling process for vulnerabilities

Metadata

URL: https://apache.org/security/committers.html

Publisher: ASF

License: ASLv2

Type: policy

Publication date: current

The ASF's Committers' Guidelines detail a comprehensive process for Apache project maintainers responding to vulnerabilities. It emphasizes confidentiality, structured routing of reports, prompt acknowledgment, private investigation, CVE identification, and coordinated patch development. The resolution phase includes internal advisory preparation and reporter collaboration, followed by synchronized public announcements of patches and advisories. Post-disclosure, the guidelines mandate website updates and CVE documentation. These detailed procedures align with international standards and legal obligations for coordinated vulnerability disclosure.

ASF vulnerability reporting process

Metadata

URL: https://apache.org/security/

Publisher: ASF

License: ASLv2

Type: policy

Publication date: current

The ASF Security Policies provide guidelines for reporting and handling vulnerabilities across Apache projects. It emphasizes confidential initial reporting, centralized tracking by the ASF Security Team, and CVE coordination. The policy clearly defines exclusions, instructs users on the correct reporting channels, and outlines a structured response including confidential handling, private fix coordination, and public advisory issuance. These policies ensure consistency in vulnerability management, aligning ASF projects with coordinated disclosure best practices and regulatory expectations, such as the EU Cyber Resilience Act.

OpenStack Foundation

OpenStack Vulnerability Management Process

Metadata

URL: https://security.openstack.org/vmt-process.html

Publisher: OpenStack Community

License: CC-BY-3.0

Type: informal

Publication date: 2011 – present

The OpenStack VMT Process describes how OpenStack manages security vulnerabilities through coordinated disclosure. It outlines the responsibilities of the independent VMT, including prompt vulnerability resolution, limiting early

information exposure, and structured handling from initial report receipt to embargoed disclosure. The process involves confidential reporting channels, internal patch review, drafting impact descriptions, and brief embargo periods. Each issue receives dedicated coordination, including CVE assignment and release scheduling. The documented practices align with industry standards for coordinated vulnerability disclosure and multi-party incident handling.

PHP Foundation

PHP Security Policies and Process

Metadata

URL: https://github.com/php/policies/blob/main/security-policies.rst

Publisher: PHP Project

License: none but CC-BY-4.0 proposed

Type: informal

Publication date: 2024 – present

PHP's Security Policies and Process is a meta-policy outlining PHP's approach to vulnerability disclosure and maintenance of public security information via the security.txt file. It directs maintainers on keeping vulnerability reporting channels clear and up-to-date, linking internal security handling with external communications. The document ensures that the security.txt is updated consistently, aligning PHP's processes with established best practices and regulatory frameworks, such as the EU Cyber Resilience Act, by providing clarity and transparency in vulnerability disclosure.

PHP Vulnerability Disclosure Policy

Metadata

URL: https://github.com/php/policies/blob/main/security-classification.rst
Publisher: PHP Project
License: none but CC-BY-4.0 proposed
Type: informal
Publication date: 2023 – present

PHP's Security Issue Classification document defines which PHP bugs constitute security vulnerabilities and categorizes them into Low, Medium, and High severity based on their potential impact and ease of exploitation. High severity issues typically allow serious compromises and receive CVE identifiers, whereas Low severity issues pose limited risks. The document outlines private handling procedures for high-risk vulnerabilities and public bug-fix approaches for lower-risk issues. It sets clear reporting guidelines and supports structured, coordinated vulnerability disclosure aligned with industry practices.

PHP Release Process

Metadata

URL: https://github.com/php/php-src/blob/master/docs/release-process.md
Publisher: PHP project
License: PHP 3.01
Type: informal
Publication date: 2007 – present

The PHP Release Process outlines detailed procedures for PHP version releases, specifically highlighting steps for security-focused updates. It distinguishes between regular updates and security releases, requiring explicit labeling and careful scheduling to optimize user response. Release managers follow structured

preparation steps including testing, packaging, and documentation, with clear guidance on announcements emphasizing the urgency of security patches. By standardizing this communication and release strategy, PHP maintains transparency and meets coordinated disclosure standards expected by compliance frameworks.

3.3 Vulnerability management guidelines

Technical Guideline TR-03183: Cyber Resilience Requirements for Manufacturers and Products - Part 1: General requirements

Metadata

URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03183/BSI-TR-03183-3-0_9_0.pdf?__blob=publicationFile&v=3

Publisher: Federal Office for Information Security (BSI), Germany

License: Not specified

Type: Technical Guideline (draft regulatory standard)

Publication date: 2024

The Technical Guideline TR-03183 from Germany's BSI delineates a set of cyber resilience requirements for manufacturers and product, aligned with the EU's upcoming Cyber Resilience Act (CRA). The guideline translates the CRA's regulatory expectations into concrete technical criteria and practices. By providing this framework in advance of the CRA's full enforcement, the document helps industry stakeholders bolster software security and compliance, ensuring that products with digital elements are developed and maintained in line with emerging

European cybersecurity standards. Part 3 describes the handling of incoming vulnerability reports.

The CERT Guide to Coordinated Vulnerability Disclosure

Metadata

URL: https://certcc.github.io/CERT-Guide-to-CVD/

Publisher: Carnegie Mellon University

License: None - approved for public release and unlimited distribution

Type: Guide

Publication date: From 2017

The CERT Guide to Coordinated Vulnerability Disclosure (CVD) is a comprehensive handbook for establishing a coordinated vulnerability disclosure process. Published by the CERT Coordination Center (part of Carnegie Mellon University's Software Engineering Institute), it walks through the entire lifecycle of handling a discovered security vulnerability when multiple parties are involved (the finder, the affected vendor, possibly intermediaries like bug bounty platforms or coordination centers). The guide covers fundamental concepts and roles (what it means to be a finder, vendor, coordinator), and lays out best practices on how to receive vulnerability reports, triage and analyze them, communicate between researchers and vendors, remediate the issues, and eventually disclose the vulnerability publicly in a responsible manner. It addresses challenges such as multi-vendor coordination (when a vulnerability affects multiple products) and dealing with disagreements or communication breakdowns. By following the CERT Guide, organizations can improve how they respond to vulnerability reports, ensuring that fixes are developed and delivered to users faster and with less conflict, ultimately reducing harm from security flaws. The original guide also exists as a PDF.

CNA Rules

Metadata

URL: https://www.cve.org/resourcessupport/allresources/cnarules

Publisher: MITRE

License: ?

Type: Semi-formal

Publication date: 2024 edition

The CNA Operational Rules are the policy framework governing how CVE Identifiers are assigned to vulnerabilities by authorized organizations. These rules, maintained by the CVE Program, detail the responsibilities of CNAs (organizations like software vendors or security coordinators that assign CVE IDs for vulnerabilities in their scope) and the processes they must follow. The document covers how CVE requests are handled, how and when a CNA should assign an ID, the required information for each CVE record, and how to communicate and publish vulnerabilities in coordination with reporters and other stakeholders. It also establishes procedures for disputes or exceptions (with the CVE Board overseeing cases that fall outside normal rules). The latest version (v4.0) of the CNA Rules introduces more flexibility and community input into CVE assignment, aiming to improve the efficiency and consistency of vulnerability identification worldwide. In essence, the CNA Rules ensure the CVE system operates smoothly, with each CNA doing its part to accurately catalog vulnerabilities in a timely manner.

EUCC Scheme Guidelines on Vulnerability Management and Disclosure

Metadata

URL: https://certification.enisa.europa.eu/document/download/5f61edd4-0151-4687-8a08-c11c103498f3_en?filename=EUCC_guidelines_vulnerability%20management%20and%20disclosure_v1.1_0.pdf

Publisher: European Union Agency for Cybersecurity (ENISA)

License: CC-BY-ND 4.0 DEED

Type: Guidelines for the EUCC Scheme

Publication date: 2025

ENISA's EUCC Vulnerability Management and Disclosure Guidelines detail requirements for vulnerability management under the EU Common Criteria certification scheme. It outlines structured internal management procedures and disclosure obligations, mandating confidential information sharing with authorities following vulnerability resolution. The guidelines ensure certified products maintain robust vulnerability management programs and compliance with EU regulatory requirements, integrating international standards into daily cybersecurity practices for manufacturers and evaluation facilities.

FIRST PSIRT Services Framework

Metadata

URL: https://www.first.org/standards/frameworks/psirts/psirt_services_framework_v1.1

Publisher: FIRST

License: Public Domain

Type: Framework

Publication date: 2015-present

The FIRST PSIRT Services Framework version 1.1 is a comprehensive reference document that describes the range of services a Product Security Incident Response Team (PSIRT) can provide. Developed by the Forum of Incident Response and Security Teams (FIRST) with input from industry experts, it enumerates and defines service categories, services, and functions specific to PSIRTs, recognizing that product-focused incident response has unique considerations compared to traditional CSIRTs. The framework covers strategic, tactical, and operational facets of a PSIRT program—from stakeholder management and vulnerability handling processes to incident coordination and post-incident analysis—offering a structured vocabulary and model for organizations either establishing a new PSIRT or benchmarking an existing one. By standardizing what effective product security response entails, the PSIRT Services Framework helps organizations ensure they meet industry best practices and regulatory expectations for managing product vulnerabilities and incidents, an area increasingly significant under product security regulations and guidelines (including parallels to the EU Cyber Resilience Act's focus on post-market security).

Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure

Metadata

URL: https://www.first.org/global/sigs/vulnerability-coordination/multiparty/guidelines-v1.1

Publisher: Forum of Incident Response and Security Teams (FIRST)

License: none

Type: Guidelines for handling coordiantion of complex (multiparty) vulnerabilities

Publication date: 2020

FIRST's Multiparty Vulnerability Coordination Guidelines provide a framework for managing vulnerabilities affecting multiple stakeholders. It addresses the complexities of contemporary software ecosystems, advocating for robust advanced planning, clear communication, minimal exposure of sensitive information, and rapid response to leaks. The guidelines include practical use-case scenarios and advice on roles, embargo timelines, and public communication strategies, complementing existing ISO standards. This structured approach ensures effective coordination in complex disclosure situations, enhancing cybersecurity resilience.

FIRST PSIRT Services Maturity Guidance

Metadata

URL: https://www.first.org/standards/frameworks/psirts/psirt_maturity_document

Publisher: FIRST

License: Public Domain

Type: Framework

Publication date: 2015-present

The FIRST PSIRT Maturity Document is a guidance framework aimed at helping organizations develop and improve their Product Security Incident Response Team capabilities over time. It builds upon the PSIRT Services Framework by outlining a maturity model—describing foundational levels and more advanced stages of PSIRT development—to illustrate how a team can evolve from basic operational readiness to a fully mature, proactive product security program. The document emphasizes essential early steps, such as securing executive sponsorship, defining clear policies (aligned with standards like ISO/IEC 29147 and 30111 for vulnerability disclosure and handling), and establishing core processes for vulnerability triage and response, before progressing to more sophisticated functions. Through this staged approach, the PSIRT Maturity guide enables organizations to assess their

current state and identify priorities for enhancement, ensuring that even as regulatory and customer expectations for robust vulnerability response increase, the organization can methodically reach higher levels of preparedness and effectiveness.

FIRST CSIRT Services Framework

Metadata

URL: https://www.first.org/standards/frameworks/csirts/csirt_services_framework_v2.1

Publisher: FIRST

License: Public Domain

Type: Framework

Publication date: 2015-present

The FIRST CSIRT Services Framework version 2.1 is a standardized taxonomy of services for Computer Security Incident Response Teams (CSIRTs), providing a structured description of the possible activities and functions such teams can perform. Co-developed by experts in the FIRST community (with support from groups like TF-CSIRT and the International Telecommunication Union), this framework lists core service categories—such as incident handling, alerts and warnings, vulnerability management, and other cybersecurity services—and breaks them down into specific sub-services and functions with clear definitions. It is intended to guide CSIRTs in defining or expanding their service portfolios, ensuring consistency in terminology and understanding across the incident response community. As the de facto reference for CSIRT operations, version 2.1 helps new and existing teams align with internationally recognized practices, which is valuable for meeting the expectations of frameworks like the EU NIS Directive/NIS2 and complementing the broader regulatory push (including the EU Cyber Resilience Act) for well-structured cybersecurity incident management capabilities.

NIST SP 800-61 Rev. 3 (Initial Public Draft): Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile

Metadata

URL: https://csrc.nist.gov/pubs/sp/800/61/r3/ipd

Publisher: NIST

License: ?

Type: informal

Publication date: 2024

This is the draft of the third revision to NIST's Computer Security Incident Handling Guide. It updates the well-known incident response lifecycle guidance (Prepare, Detect and Analyze, Contain, Eradicate and Recover, Post-Incident) to address modern challenges and to align with the upcoming NIST Cybersecurity Framework 2.0. The draft emphasizes integrating incident response with overall risk management and uses the CSF structure to organize recommendations. It highlights considerations such as cloud incidents, supply-chain incidents, and coordinating with external entities (like law enforcement or industry sharing groups) – topics that have grown in importance since the previous Rev. 2. The draft also suggests metrics and continuous improvement practices for incident response teams, encouraging organizations to not just react to incidents but to collect lessons learned and feed those back into defensive measures. In essence, SP 800-61 Rev. 3 will serve as a contemporary playbook for cybersecurity incident response, ensuring teams are well-prepared and that their processes fit into a broader strategy for resilience.

Guide to coordinated vulnerability disclosure for open source software projects

Metadata

URL: https://github.com/ossf/oss-vulnerability-guide

Publisher: OpenSSF

License: CC-BY-4.0

Type: Guidance, templates, and advise for how open source projects and security researchers can better coordinate vulnerability disclosures together

Publication date: 2022 – present

The Open Source Security Foundation's (OpenSSF) Guide to Coordinated Vulnerability Disclosure educates open-source maintainers about structured vulnerability management. It describes the coordinated disclosure lifecycle, from preparation and initial reporting through confidential assessment, remediation under embargo, and public disclosure. The guide provides practical templates, emphasizes proactive planning, and is informed by industry best practices. This resource supports the implementation of consistent disclosure practices in open-source communities, assisting maintainers in meeting regulatory demands and improving overall software security maturity.

Patching and Updates Guidelines from Berkeley Information Security Office

Metadata

URL: https://security.berkeley.edu/MSSND/patching-and-updates-guidelines

Publisher: UC Berkeley

License:

Type: informal

Publication date:

Published by the University of California, Berkeley's Information Security Office, this guideline provides best practices for keeping systems and applications up to date with security patches. It stresses the importance of timely patching as a critical defense against exploits of known vulnerabilities. Key recommendations include maintaining an inventory of all IT assets to know what needs patching, categorizing updates by severity and applying critical security patches as soon as possible (ideally within a defined timeframe like 48-72 hours for high severity), and establishing a regular patch cycle for less urgent updates. The guidelines also suggest testing patches in a staging environment when feasible, having a rollback plan in case an update causes issues, and using automated management tools to deploy patches at scale across an organization's devices. Additionally, the document might cover how to handle systems that can't be patched (due to compatibility or support issues), such as isolating them or applying compensating controls. Overall, the Berkeley guidelines serve as a practical manual for IT teams to build a robust patch management process, which in turn reduces an organization's exposure to known exploits.

4. SBOM

This section contains references which are relevant to the requirements expressed in Annex I, Part II(1) of the CRA. It corresponds to the horizontal type "B" standard number 15 of the European Commission's standardisation request to the ESOs:

> *Manufacturers of products with digital elements shall: (1) identify and document vulnerabilities and components contained in products with digital elements, including by drawing up a software bill of materials in a commonly used and machine-readable format covering at the very least the top-level dependencies of the products;*

4.1 Technical specifications for SBOMs

ECMA-424

Metadata

URL: https://ecma-international.org/publications-and-standards/standards/ecma-424/

Publisher: ECMA International

License: ?

Type: Formal

Publication date: June 2024 (version 1.0)

CycloneDX, originally a BOM (Bill of Materials) format from OWASP for software components, has been ratified as an international standard by Ecma International (as ECMA-424). This standard formalizes the schema and usage of CycloneDX for creating Software Bills of Materials, which enumerate the components and

dependencies in a software product. ECMA-424 (CycloneDX v1.6) defines how to represent components, their licenses, version information, relationships, and relevant metadata (like hashes, integrity proofs, and known vulnerabilities) in a machine-readable way. The publication of CycloneDX as ECMA-424 means it is a globally recognized specification, facilitating widespread adoption. It enables organizations to exchange SBOM information consistently across tools and industries, enhancing transparency and helping stakeholders (developers, consumers, regulators) to track and manage supply chain risk by knowing exactly what is inside a software product.

ISO/IEC 5962:2021

Metadata

URL: https://www.iso.org/standard/81870.html

Publisher: ISO

License: ?

Type: Formal

Publication date: August 2021 (Version 1.0)

ISO/IEC 5962:2021 is the international standardization of the Software Package Data Exchange (SPDX) format as a norm for Software Bill of Materials. SPDX originated as a Linux Foundation project to describe the components and licenses in software, and it was brought into ISO to encourage global adoption. The standard defines how to list all pieces of software (open-source libraries, proprietary modules, etc.), along with metadata like their versions, checksums, licenses, and relationships, in a structured document. Using ISO 5962 (SPDX), organizations can share SBOMs across organizational and tool boundaries with confidence that everyone interprets the content the same way. This promotes transparency in software supply chains, helps with automated license compliance checks, and aids vulnerability management (by quickly identifying if a product

contains a component with a known CVE). SPDX as an ISO standard underscores its maturity and the importance of SBOMs in modern cybersecurity and compliance workflows.

4.2 Technical specifications for software identification

Software Identification for Cybersecurity: Survey and Recommendations for Regulators

Metadata

URL: https://swhsec.github.io/pdf/swhid_for_cybersecurity_regulations.pdf

Publisher: Olivier Barais, Roberto Di Cosmo, Ludovic Mé, Stefano Zacchiroli, and Olivier Zendra

License: Not specified

Type: Technical report (survey & policy recommendations)

Publication date: 2025

This paper provides a comprehensive overview of how software components—especially open-source elements—can be uniquely identified to improve cybersecurity. It examines current challenges in software identification and reviews existing identifier schemes, highlighting issues like naming inconsistencies and ephemerality. In response, the authors advocate for Software Heritage persistent IDs (SWHIDs), which are content-derived, permanent identifiers, as a unified solution for tracking software artifacts across systems. The report emphasizes that SWHIDs would facilitate better vulnerability management and transparency, aligning with emerging regulatory requirements (such as the EU Cyber Resilience

Act and U.S. executive orders) by ensuring every component in the supply chain can be traced and verified in a stable, tool-agnostic manner.

Unique identifier for software components

Metadata

URL: https://github.com/package-url/purl-spec

Publisher: ECMA International (in 2025)

License: ?

Type: Formal

Publication date: 2025

The Package URL specification defines a standard way to identify software packages across different programming ecosystems using a simple, URL-like format. A “purl” takes the form of a URI that includes a package’s type (e.g. npm, Maven, PyPI, GitHub), the namespace or group (if applicable), the package name, version, and any qualifiers or subpath. For example, a purl might look like: pkg:maven/org.apache.commons/commons-lang3@3.12.0. This one identifier concisely tells you the package manager (Maven), group (org.apache.commons), artifact (commons-lang3), and version (3.12.0). The purl spec, maintained on GitHub, has become widely used in tools and SBOM formats because it provides a consistent way to refer to components. It helps automate the tracking of dependencies, vulnerability matching (linking known vulnerabilities to package coordinates), and license compliance, since every package instance can be referenced uniformly. In summary, Package URL is like a universal addressing system for software components, making it easier to integrate data about packages across different systems and databases.

Unique identifier for software components

Metadata

URL: https://www.swhid.org/specification

Publisher: Software Heritage (ISO pending)

License: ? (Issue raised on GitHub)

Type: Semi-formal

Publication date: Dec 2023 (v 1.1)

The Software Heritage ID is a scheme for assigning permanent, unique identifiers to software artifacts (such as source code files, commits, or releases) based on their content. The specification (developed by the Software Heritage archive) defines a cryptographic hash-based ID format that includes a prefix (swh:), a content hash, and additional qualifiers to denote the type of artifact and its position in the archive's Merkle graph. For example, there are SWHIDs for files (blobs), directories (which list files), commits (snapshots of directories with metadata), and releases (tags). Because SWHIDs are content-derived, they are immutable and globally unique: the same piece of code will always have the same SWHID. This system enables precise references to software components regardless of where they are stored or how they are named elsewhere. In practice, SWHIDs facilitate long-term preservation, citation, and tracking of software, since one can resolve an SWHID via the Software Heritage archive to retrieve the exact artifact, and they help with supply chain integrity by ensuring that references to source code are unambiguous and verifiable.

4.3 SBOM implementation guidelines

Technical Guideline TR-03183: Cyber Resilience Requirements for Manufacturers and Products - Part 2 Software Bill of

Materials (SBOM)

Metadata

URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03183/BSI-TR-03183-2-2_0_0.pdf?__blob=publicationFile&v=3

Publisher: Federal Office for Information Security (BSI), Germany

License: Not specified

Type: Technical Guideline (draft regulatory standard)

Publication date: 2024

The Technical Guideline TR-03183 from Germany's BSI delineates a set of cyber resilience requirements for manufacturers and product, aligned with the EU's upcoming Cyber Resilience Act (CRA). The guideline translates the CRA's regulatory expectations into concrete technical criteria and practices. By providing this framework in advance of the CRA's full enforcement, the document helps industry stakeholders bolster software security and compliance, ensuring that products with digital elements are developed and maintained in line with emerging European cybersecurity standards. Part 2 describes formal and technical requirements for Software Bill of Materials (SBOM).

Detailed Software Supply Chain Uses Cases for SCITT

Metadata

URL: https://datatracker.ietf.org/doc/draft-ietf-scitt-software-use-cases/

Publisher: IETF

License:

Type: Internet-Draft

Publication date: 2024

The IETF's SCITT working group is developing a set of protocols to enable a trustworthy ledger of software supply chain events (like signing, transferring, and auditing software components). The linked document on software use cases describes scenarios where organizations need a verifiable record of actions in the supply chain. For example, one use case might be tracking the provenance of a piece of code: a developer commits code and signs it, a build service records that it built a binary from specific source, a scanner logs that it scanned the binary for vulnerabilities, etc. SCITT envisions a tamper-evident ledger (possibly blockchain-like or using transparency logs) where each of these actions is recorded by actors with cryptographic signatures. The use case document illustrates how this can help – for instance, a consumer of software can query the ledger to see if a given software package version has a recorded build attestation and security scan results before trusting it. Another use case could involve regulatory compliance: proving to an auditor that every software component in a device went through certain security checks. By enumerating such cases, the document guides the design of the SCITT architecture to ensure it meets real-world needs. In essence, SCITT's use cases highlight the forthcoming ability to establish a chain of trust for software artifacts through standardized evidence, improving the integrity and transparency of the software supply chain.

JavaScript SBOM and Software Attestation Challenges and Recommendations

Metadata

URL: https://github.com/openjs-foundation/security-collab-space/blob/main/OpenJS-SBOM-CSCRM-Challenges-Recommendations.md

Publisher: OpenJS Security Collaboration Space

License:

Type: informal

Publication date: 2024

This resource is a collaborative report produced by the OpenJS Foundation's security working group, addressing the practical challenges of implementing Software Bill of Materials (SBOM) and Cyber Supply Chain Risk Management (C-SCRM) in open source projects. It identifies key obstacles, such as generating comprehensive SBOMs, ensuring their accuracy and maintenance, and integrating supply chain security practices into development workflows, especially within widely-used JavaScript ecosystems. The document offers recommendations to mitigate these issues, advocating for improved tooling, standardized processes, and cross-project knowledge sharing to strengthen supply chain transparency and vulnerability management. By highlighting these challenges and solutions, the report provides guidance relevant to improving software security posture in line with emerging regulatory expectations (like those of the EU Cyber Resilience Act) that emphasize software supply chain integrity and documentation.

Energy Central article

Metadata

URL: https://energycentral.com/c/iu/facts-and-opinions-about-sbom-implementation-and-adoption

Publisher: Energy Central/Dick Brooks

License:

Type: informal

Publication date: Febrary 16, 2025

This article, written by a practitioner with extensive experience in SBOMs, shares practical insights and opinions on the current state of SBOM usage. One key point the author makes is that the only authoritative SBOM for a product should come from its original producer – they have the best knowledge of the components. However, even producers might not have perfect insight into every sub-component, especially when they incorporate third-party elements, highlighting the need for upstream component transparency. The author argues against waiting for SBOM technology to be “perfect” or universally adopted; even imperfect SBOMs provide significant value right now by giving visibility into software contents and risks. The article notes that many tools and a vibrant community already exist to help generate and use SBOMs, debunking the myth that SBOM is immature. It also reminds readers that there are two well-established SBOM standards (CycloneDX and SPDX) that are sufficient for current needs. The author cautions that there are both proponents and detractors of SBOMs speaking in the industry, and suggests readers do their own evaluations – in other words, experiment with SBOMs on their own software to see the benefits firsthand. Overall, the piece encourages organizations to start integrating SBOM practices into their workflows today, to improve software risk management and comply with emerging requirements, rather than adopting a “wait and see” approach.

The Minimum Elements for a Software Bill of Materials (SBOM)

Metadata

URL: https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf
Publisher: United States Department of Commerce (NTIA)
License: Public Domain
Type: Government report (cybersecurity guidance)
Publication date: 2021

This official report outlines the fundamental requirements for a minimum Software Bill of Materials (SBOM) as part of a national cybersecurity initiative. It specifies the core data elements that an SBOM should contain (such as component names, versions, and suppliers), along with the need for machine-readable formats and automated tooling to generate and use SBOMs at scale. The document also describes basic organizational processes for maintaining SBOMs and leveraging them in managing software vulnerabilities and license compliance. Issued under the directive of a U.S. Executive Order aimed at improving cybersecurity, the report establishes a common baseline for SBOM content and practices, laying a foundation for greater transparency in software supply chains while noting potential extensions for future use cases beyond the minimum requirements.

Technical guidelines on SBOMs

Metadata

URL: https://www.cert-in.org.in/PDF/SBOM_Guidelines.pdf

Publisher: CERT-IN

License:

Type: informal

Publication date: 03.10.2024 (Version 1.0)

In October 2024, the Indian Computer Emergency Response Team (CERT-In) released a set of technical guidelines on Software Bill of Materials (SBOM) for organizations in the public sector and critical industries. This document underscores the value of SBOMs as a tool for transparency in software supply chains and provides recommendations for how to generate and maintain SBOMs. It outlines processes and best practices for implementing SBOMs – such as what component information should be included (name, version, supplier, known vulnerabilities, etc.) – and offers guidance on integrating SBOM use into procurement and risk management. The goal of these guidelines is to ensure that Indian government agencies and essential service providers adopt SBOM practices to enhance software security and resilience, in line with global trends and regulatory expectations.

Technical guidelines on SBOMs

Metadata

URL: https://www.cisa.gov/sites/default/files/2024-07/PDM24050%20Software%20Acquisition%20Guide%20for%20Government%20Enterprise%20ConsumersV2_508c.pdf

Publisher: US Government

License:

Type: formal

Publication date: 2024

In mid-2024, CISA released updated guidance (version 2.0 of its Software Acquisition guidance for enterprises) focusing on how government organizations should handle SBOMs. This technical guide provides detailed procedures for requesting SBOMs from software vendors, ingesting and validating those SBOMs, and using them for risk analysis. It likely advises agencies to use automated tools to parse SBOM data to identify components and check them against vulnerability databases (so if a vendor provides an SBOM, the agency can quickly see if any included library has known flaws). It also covers establishing workflows to manage SBOM data at scale – for example, keeping a repository of SBOMs for all software in use, updating it when software is patched or upgraded, and integrating SBOM review into continuous monitoring. The guide emphasizes that simply obtaining SBOMs is not the end goal; organizations must also develop the capability to act on SBOM information (like prioritizing patches or isolating systems when a critical vulnerability in a common component emerges). By following these guidelines, government enterprises can move toward proactive supply chain security management, gaining visibility into their software assets and being better prepared to respond to new threats or compliance checks (like those required by EO mandates).

5. Due diligence requirements

This section contains references which are relevant to the requirement to "exercise due diligence" expressed in Article 13(5) of the CRA. It has no corresponding standard request in the European Commission's standardisation request to the ESOs.

> *For the purpose of complying with paragraph 1, manufacturers shall exercise due diligence when integrating components sourced from third parties so that those components do not compromise the cybersecurity of the product with digital elements, including when integrating components of free and open-source software that have not been made available on the market in the course of a commercial activity.*

US CISA Software Acquisition Guide

Metadata

URL: https://cisa.gov/sag

Publisher: CISA

License:

Type:

Publication date: 2024

The U.S. CISA Software Acquisition Guide provides practical guidance to government agencies (and large enterprises) on how to incorporate security considerations into the procurement of software. It outlines how buyers should define security requirements in RFPs/contracts – such as requiring vendors to follow secure development standards (like the NIST SSDF), provide artifacts like SBOMs and vulnerability disclosure policies, and agree to notify and patch if vulnerabilities are found. It also advises on evaluating supplier risk (e.g. checking if

the vendor has had frequent security issues historically or if they have certified processes like ISO 27001). The guide suggests structuring the vendor selection process to include security questionnaires or assessments and to prefer vendors who can demonstrate secure-by-design approaches (perhaps via certifications or past performance). Additionally, it covers how to handle acquired software: establishing processes for incoming SBOM analysis, periodic re-scans for vulnerabilities in the product, and ensuring maintenance contracts cover security updates. By following this guide, procurement officers and risk managers can make more informed decisions, selecting products that not only meet functionality needs but also contribute positively to the organization's security posture, thereby using market forces to encourage vendors to deliver safer software.

Good Practices in Supply Chain Cybersecurity

Metadata

URL: https://www.enisa.europa.eu/sites/default/files/publications/Good%20Practices%20for%20Supply%20Chain%20Cybersecurity.pdf

Publisher: ENISA

License: CC-BY-4.0

Type:

Publication date: 2023 – present

This report by the European Union Agency for Cybersecurity (ENISA) surveys the threat landscape of ICT supply chains and offers recommendations to secure them. It identifies common attack scenarios and weaknesses in digital supply chains (such as compromise of third-party software dependencies, insecure equipment from suppliers, or poor internal supplier security processes). The document then presents a set of good practices for different stakeholders – including manufacturers, service providers, and customers – to mitigate these risks. These practices range from performing due diligence and security audits on suppliers,

incorporating security requirements into procurement contracts, ensuring suppliers have vulnerability management and incident response procedures, to establishing multi-tier trust (so that suppliers enforce similar standards on their subcontractors). By following these practices, organizations can strengthen the weakest links in their supply chain and reduce the risk of cascading security incidents originating from outside partners or components.

US NASA procurement practices and process

Metadata

URL: https://www.nasa.gov/secure-software-development-self-attestation-resources-and-knowledge/

Publisher: NASA

License:

Type:

Publication date: 2024

In response to federal requirements (stemming from Executive Order 14028) that government agencies only use software following secure development practices, NASA has instituted a process where software suppliers must self-attest to their security. NASA's public resources for this include guidance and forms for vendors. Essentially, a contractor providing software to NASA needs to review their own software development processes against criteria (largely drawn from NIST's Secure Software Development Framework and related guidance) and then sign a statement confirming they comply. The NASA site likely provides a checklist or template covering topics like: does the vendor conduct threat modeling? Do they run static code analysis? How do they manage third-party component vulnerabilities? etc. It might also have educational material to help suppliers understand what practices are expected. By implementing this attestation process, NASA ensures its contractors are consciously evaluating and improving their security postures. This ultimately reduces risk in NASA's supply chain—only software built with a baseline

of security best practices should be running in NASA environments—and it reinforces industry adoption of secure development norms.

OpenChain ISO/IEC 18974 - Security Assurance

Metadata

URL: https://openchainproject.org/security-assurance

Publisher: OpenChain/ISO

License: CC-BY-4.0

Type:

Publication date: 45261

ISO/IEC 18974:2023, also known as the OpenChain Security Assurance standard, defines the key requirements for establishing a quality open source software security assurance program. It is designed to help organizations systematically identify and address known vulnerabilities in the open source components they use (for example, by checking for CVEs and dependency alerts). The standard outlines where in the software supply chain security processes should be in place, how roles and responsibilities for security should be assigned, and how to sustain these processes over time. It is intended to be lightweight and accessible, with support from the OpenChain community (providing reference materials and self-certification checklists), so that organizations of all sizes can adopt it and certify that their open source handling meets a baseline of security trustworthiness.

6. Security attestations

This section contains references which are relevant to the "security attestation of free and open-source software" described in Article 25 of the CRA. It has no corresponding standard request in the European Commission's standardisation request to the ESOs.

In order to facilitate the due diligence obligation set out in Article 13(5), in particular as regards manufacturers that integrate free and open-source software components in their products with digital elements, the Commission is empowered to adopt delegated acts in accordance with Article 61 to supplement this Regulation by establishing voluntary security attestation programmes allowing the developers or users of products with digital elements qualifying as free and open-source software as well as other third parties to assess the conformity of such products with all or certain essential cybersecurity requirements or other obligations laid down in this Regulation.

Authoritative Guide to Attestations

Metadata

URL: https://cyclonedx.org/guides/OWASP_CycloneDX-Authoritative-Guide-to-Attestations-en.pdf

Publisher: OWASP

License: CC-BY-SA 4.0

Type: informal

Publication date: 2024 – present

This guide provides organizations with a framework for digitally transforming their audit and attestation workflows using standardized, machine-readable attestations.

An attestation in this context is a cryptographically signed statement about some aspect of a software product or process (for example, an attestation that a product was built in a secure environment or complies with certain standards). The guide explains how to create and use such attestations, leveraging the CycloneDX format, to document compliance or security assurances in an automated way. By following the guide, organizations can move away from purely manual audit statements and instead generate verifiable digital evidence of their security practices, making it easier to trust and verify software in supply chains or regulatory contexts.

Best Practices Badge

Metadata

URL: https://www.bestpractices.dev/en

Publisher: OpenSSF

License: MIT/CDLA-Permissive

Type:

Publication date: from 2021 under the current name

The OpenSSF Best Practices Badge Program is a voluntary self-certification initiative that allows open-source software projects to demonstrate their adherence to a broad range of security and quality best practices. Projects complete a web-based questionnaire covering topics like version control, vulnerability disclosure, testing, code review, and build process hygiene. If they meet the criteria, they earn a badge that can be displayed to indicate the project follows industry-recommended practices for open source development. The program (originating from the Core Infrastructure Initiative and now under the Open Source Security Foundation) aims to improve software health and transparency; consumers of open source can use the badge as one indicator that a project is more likely to produce secure, reliable software.

FreeBSD SSDF Attestation

Metadata

URL: https://freebsdfoundation.org/news-and-events/latest-news/freebsd-foundation-announces-ssdf-attestation/
Publisher: FreeBSD Foundation
License: Confidential
Type: attestation
Publication date: 45599

In November 2023, the FreeBSD Foundation announced the availability of a Secure Software Development Framework (SSDF) Attestation, a formal report that documents how the FreeBSD open-source operating system's development practices align with NIST's SSDF guidelines. This attestation was developed to assist commercial users of FreeBSD in meeting emerging security requirements—specifically, U.S. government software procurement rules that require suppliers to self-attest to following secure development practices (as mandated by NIST SP 800-218 and related OMB directives). The announcement highlights FreeBSD's longstanding emphasis on security by design and explains that the attestation service allows vendors and cloud providers relying on FreeBSD to easily demonstrate that the upstream software meets recognized secure development criteria. By providing this attestation, the FreeBSD Foundation bridges open-source development with regulatory compliance needs, reflecting broader trends (in the U.S. and potentially in the EU through initiatives like the Cyber Resilience Act) to demand greater assurance of software supply chain security.

OpenJS Ecosystem Sustainability Program (ESP)

Metadata

URL: https://openjsf.org/blog/ecosystem-sustainability-program

Publisher: OpenJS

License:

Type: informal

Publication date: 2024

The OpenJS Ecosystem Sustainability Program (ESP) is an initiative by the OpenJS Foundation aimed at enhancing the security and longevity of widely-used but under-maintained JavaScript projects. Launched with HeroDevs as its inaugural partner, the program offers long-term support for legacy software, exemplified by the introduction of "Never-Ending Support" (NES) for Express.js. ESP is an opt-in, partner-driven model designed to secure outdated software and generate new revenue streams for OpenJS projects. By providing structured support for aging yet critical components of the JavaScript ecosystem, ESP addresses software supply chain risks and aligns with regulatory expectations for secure software maintenance, such as those outlined in the EU Cyber Resilience Act.

Secure Software Development Attestation Form

Metadata

URL: https://www.cisa.gov/resources-tools/resources/secure-software-development-attestation-form

Publisher: CISA

License: Public Domain

Type: Form

Publication date: 2024

In March 2024, the Cybersecurity and Infrastructure Security Agency (CISA) and the U.S. Office of Management and Budget (OMB) jointly released the Secure Software Development Attestation Form. This document serves as a compliance mechanism

under federal cybersecurity initiatives, requiring software producers to affirm adherence to National Institute of Standards and Technology (NIST) secure software development guidelines in accordance with Executive Order 14028 and OMB memoranda M-22-18 and M-23-16. The form's purpose is to provide assurance that any software used by federal agencies is securely developed, effectively enforcing minimum secure development standards and toolsets among vendors. By tying this attestation to U.S. federal procurement, the government compels software suppliers to demonstrate compliance with specified development practices, and failure to provide a signed attestation may result in agencies discontinuing use of the software. Overall, the resource establishes a uniform attestation process that supports regulatory compliance and strengthens the security of the software supply chain in government acquisitions.

SLSA Supply Chain Levels for Software Artifacts

Metadata

URL: https://github.com/slsa-framework/slsa

Publisher: OpenSSF

License: Apache 2.0 and others

Type: Semi-formal

Publication date:

SLSA (pronounced "salsa") is a security framework for improving software supply chain integrity, created by a consortium of industry leaders. It defines a maturity model with four levels that software producers can achieve, each level adding more advanced supply chain protections. At Level 1, basic requirements like using version control and build scripts are in place. By Level 2, tamper-evident builds are required (e.g. generating provenance metadata). Level 3 mandates trusted, verifiable builds (such as using a dedicated, isolated build service to prevent interference), and Level 4 involves the highest assurances including two-party review of all changes and a hermetic, reproducible build process. The framework is

documented openly (on GitHub) and provides specifications for how to produce attestations (metadata) that a build meets a certain SLSA level. The purpose of SLSA is to thwart supply chain attacks (like code tampering, compromised dependencies, etc.) by incrementally hardening the build and release process. Organizations can adopt SLSA guidelines to progressively make their software build pipelines more secure, and customers can eventually prefer software that comes with SLSA provenance, indicating it was built under strict controls.

7. Similar legislation

This section contains references to legislation that is similar or related to the CRA. It is organized by country.

Australia

Australia Code of Practice - Securing the Internet of Things for Consumers

Metadata

Publisher:

License:

Type:

Publication date:

Australia released a voluntary Code of Practice for consumer IoT security in 2020 (developed by the Department of Home Affairs). This code closely mirrors the UK's 13 IoT security principles and ETSI's guidelines. It includes practices such as: don't use default passwords (and preferably implement unique passwords per device), implement a vulnerability disclosure policy, keep software updated and securely updateable, securely store credentials, minimize data collection, and ensure devices can be securely deleted or reset by users. The Australian Code of Practice serves as guidance for industry; while not legally binding, it was endorsed by the government as the expected norm. Companies are encouraged to adopt it and even use it as a competitive advantage (demonstrating compliance could be marketed to consumers). The code also set the stage for potential future regulation if voluntary uptake is insufficient. By articulating clear best practices, Australia's government aimed to elevate IoT security and protect Australian consumers from common IoT threats like unauthorized access or data breaches via their smart

devices. The principles from this code of practice have since influenced discussions on IoT security regulation and labeling in Australia (such as the work towards an IoT security “star rating” similar to other countries).

Finland

Finland’s national consumer IoT certification scheme

Metadata

Publisher:

License:

Type:

Publication date:

- Finland was a pioneer in Europe for consumer IoT security certification, launching its cybersecurity label program in 2019. In this scheme run by Traficom (the Finnish Transport and Communications Agency), manufacturers can apply to have their smart products evaluated against a set of security criteria. Certified products receive a “Cybersecurity Label” (Tietoturvamerkki in Finnish) that they can display to inform consumers that the device meets Finland’s security requirements. The criteria include things like no default passwords, secure communications, and a commitment by the vendor to update the product’s software to fix vulnerabilities. The Finnish label is voluntary but has seen adoption in products like smart watches, home hubs, and IoT appliances. The program’s goal is to raise consumer awareness and reward companies that invest in security. Finland’s label has gained international recognition – it has mutual recognition with Singapore’s CLS (whereby a product meeting the Finnish label is accepted as Level 3 in Singapore’s scheme). This national certification helps build trust in IoT products and influenced the development of similar labeling efforts across the EU.

Germany

Germany BSI IoT label

Metadata

URL: https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/IT-Sicherheitskennzeichen/it-sicherheitskennzeichen_node.html

Publisher:

License:

Type:

Publication date:

- The German Federal Office for Information Security (BSI) introduced an “IT-Sicherheitskennzeichen” (IT Security Label) in 2022 for consumer IT products. This is a voluntary label that manufacturers can obtain to signal that their product meets basic cybersecurity requirements and that the manufacturer is transparent about its security features. Rather than extensive lab testing, the BSI label currently works as a self-declaration: the manufacturer commits to certain practices (for example, providing software updates for a minimum period, and having no universal default passwords) and BSI publishes a webpage for each labeled product detailing its security properties and update policy. The label itself is often a QR code that consumers can scan to read those details on BSI’s website. This approach educates consumers on what the device does for security and what they as users should do (like applying updates). Products like routers, smart TVs, and cameras have started getting the label. The label program also ties into international efforts: Germany and Singapore have a mutual recognition arrangement (devices with Germany’s label count as at least CLS Level 2 in Singapore and vice versa). Overall, the BSI IT Security Label aims to increase IoT product security through transparency and by pushing manufacturers to adhere to baseline good practices in order to earn the official recognition.

India

CSCRF for SEBI REs

Metadata

URL: https://www.sebi.gov.in/legal/circulars/aug-2024/cybersecurity-and-cyber-resilience-framework-cscrf-for-sebi-regulated-entities-res-_85964.html

Publisher: SEBI

License:

Type: Formal

Publication date: 45524

Introduced by the Securities and Exchange Board of India in August 2024, the CSCRF is a comprehensive framework of cybersecurity guidelines for India's capital market entities (stock exchanges, depositories, brokers, asset managers, etc.). It mandates that these regulated entities implement uniform baseline measures to anticipate cyber threats, protect systems, detect incidents, respond to attacks, and recover operations – aligning with broader cyber resilience goals. The CSCRF sets out objectives like addressing evolving threats, aligning with international standards, enforcing regular cybersecurity audits, and standardizing incident reporting. It introduces structured requirements by combining Cyber Resilience Goals (Anticipate, Withstand, Recover, Evolve) with Cybersecurity Functions (such as Governance, Identify, Protect, Detect, Respond). Entities are categorized by size/impact and must implement controls appropriate to their category. Overall, this framework elevates the cybersecurity posture of India's financial sector by requiring consistent practices and accountability across all regulated organizations.

India TEC Code of Practice for Securing Consumer IoT

Metadata

Publisher:

License:

Type:

Publication date:

India's Telecommunication Engineering Centre (TEC), under the Department of Telecom, released a "Code of Practice" in January 2022 as a guideline for consumer IoT security. This code of practice is largely aligned with global benchmarks like ETSI EN 303 645 and the UK's 13 principles for IoT security. It provides a list of best practices that manufacturers should implement: some examples include using unique per-device passwords, implementing a vulnerability disclosure mechanism, securing personal data, ensuring communications are encrypted, and providing software updates to devices. While not a regulation, this document serves as an official reference for what the Indian government expects in terms of IoT device security. It's intended for device manufacturers, IoT service providers, and app developers to consult and adopt these measures in their products and services. By following the TEC's code, IoT companies can significantly raise the security level of their products in India, and it prepares them for any future mandatory requirements. The issuance of this guideline reflects India's recognition of the growing risks posed by the proliferation of IoT devices and the need to protect consumers and critical networks from those risks.

United Kingdom

U.K.'s Product Security and Telecommunications Infrastructure Bill

Metadata

Publisher:

License:

Type:

Publication date:

The UK PSTI Act 2022 is a landmark law that establishes cybersecurity requirements for consumer connectable products (the IoT devices). Building on prior voluntary guidelines, it makes certain practices legally mandatory for manufacturers, importers, and distributors in the UK. The Act's first set of requirements (which will be enforced via secondary regulations) include: banning universal default passwords in devices, requiring a public point of contact for vulnerability reporting, and mandating transparency about the minimum time period during which the device will receive security updates. Manufacturers must provide a compliance statement for their products covering these aspects. The law gives regulators power to fine companies that don't comply. Essentially, PSTI turns basic IoT security principles into law – for example, if a toy or appliance ships with a default password like "admin" that is not unique, that will be illegal to sell in the UK. The Act also gives the government flexibility to expand requirements over time (it can add further provisions like mandates around data encryption or firewalls if needed). By legislating these measures, the UK aims to remove the most egregious vulnerabilities from consumer devices and drive manufacturers globally to improve the security of the products that end up in British households.

United States of America

CISA Secure by Design Pledge

Metadata

URL: https://www.cisa.gov/sites/default/files/2024-05/CISA%20Secure%20by%20Design%20Pledge_508c.pdf

Publisher: CISA

License:

Type:

Publication date: 2024

The Secure by Design pledge is an initiative by the U.S. Cybersecurity and Infrastructure Security Agency calling on technology companies to fundamentally shift their development philosophy to prioritize security from the start. The pledge outlines core principles such as developing software with a proactive adversary mindset (anticipating how software could be misused or attacked), ensuring default configurations are secure out-of-the-box (so customers don't have to harden products themselves), practicing transparency in vulnerability handling (including publishing SBOMs and disclosing vulnerabilities), and enabling continuous security testing and feedback during development. Companies that take the pledge publicly commit to these values, signaling that they will invest in practices like threat modeling, secure coding training, rigorous testing, and building security features (like authentication and encryption) as integral parts of their products. The broader goal is cultural: to move the industry away from treating security as an afterthought or add-on, and instead have security as a foundational aspect of product design, thereby reducing the prevalence of easily exploitable weaknesses in widely used technology.

Cybersecurity Labelling Scheme for IoT - CLS(IoT)

Metadata

URL: https://www.csa.gov.sg/our-programmes/certification-and-labelling-schemes/cybersecurity-labelling-scheme/about/

Publisher: CSA

License:

Type: Formal

Publication date:

Singapore's Cybersecurity Labelling Scheme is a four-tier rating program launched in 2020 by the Cyber Security Agency (CSA) to improve the security of consumer smart devices. Under CLS, IoT products (like home routers, smart cameras, etc.) are evaluated and given a security grade of Level 1 (basic) to Level 4 (most robust), indicated by the number of stars on the label. Each level corresponds to a set of cybersecurity provisions: for example, Level 1 requires adherence to baseline security requirements (largely based on ETSI EN 303 645), higher levels add requirements like using a vetted standard (Level 2), conducting structured lifecycle security measures and binary analysis (Level 3), and perhaps formal certification or penetration testing (Level 4). The label helps consumers easily identify products with better cybersecurity and incentivizes manufacturers to incorporate stronger security into design to achieve higher ratings. Singapore's CLS is one of the first schemes of its kind in Asia and includes mutual recognition arrangements (e.g., a device meeting Finland's cybersecurity label is recognized at CLS Level 3, and vice versa), promoting international coherence in IoT security standards.

CyberTrust Mark

Metadata

URL: https://www.fcc.gov/CyberTrustMark

Publisher: US FCC

License:

Type: formal

Publication date: 2023

The Cyber Trust Mark is a forthcoming labeling scheme introduced by the U.S. Federal Communications Commission (FCC) to signify consumer IoT products that meet certain cybersecurity benchmarks. Under this program, makers of smart devices (like smart thermostats, fitness trackers, connected appliances, etc.) can voluntarily undergo an evaluation of their product's security features. The baseline criteria likely include things like: unique default passwords or user setup of credentials, a commitment to provide security updates for a minimum period, having a vulnerability reporting mechanism, and perhaps compliance with standards such as ETSI EN 303 645. Devices that pass the evaluation will be allowed to display the "Cyber Trust Mark" logo (possibly a shield icon or similar) on their packaging. This functions similarly to an EnergyStar label but for cybersecurity. The idea is to help consumers easily identify products that are deemed more secure, thereby encouraging manufacturers to improve security to earn the mark. Over time, widespread adoption of the Cyber Trust Mark aims to raise the minimum security level of IoT devices available in the U.S. market by combining consumer choice with a recognizable seal of approval for security.

EO 14144

Metadata

URL: https://www.federalregister.gov/documents/2025/01/17/2025-01470/strengthening-and-promoting-innovation-in-the-nations-cybersecurity#p-10

Publisher: US Government

License:

Type:

Publication date: 2025

A key aspect of Executive Order 14144 is its directive to federal procurement authorities to define and enforce criteria for “secure and trustworthy” products. The Executive Order mandates that federal agencies, when buying software or connected devices, must ensure those products meet rigorous cybersecurity standards – effectively using government contracts as a driver for better security. It tasks NIST and other agencies with establishing what technical evidence or certifications a vendor must provide to be considered trustworthy. This could include providing an SBOM for supplied software, proving that the software was developed under secure processes (via self-attestation or third-party audit), demonstrating compliance with specific standards (like FIPS-validated cryptography, or adherence to zero-trust principles if relevant), and ensuring mechanisms for swift vulnerability remediation. The Federal Acquisition Regulation (FAR) will be updated accordingly, making these requirements legally binding in contracts. In practical terms, this means a software company wanting to sell to the U.S. government will need to implement robust security in their product and supply chain or risk being excluded. The intent is both to protect government systems from supply chain attacks and to influence the broader market: when large vendors improve security to sell to the government, those improvements often propagate to all customers.

Vietnam

Vietnam's Cyber Information Security Requirements for Internet of things

Metadata

Publisher:

License:

Type:

Publication date:

In 2021, Vietnam's Ministry of Information and Communications issued a set of baseline cybersecurity requirements for consumer Internet-of-Things devices (for example, network cameras, Wi-Fi routers, and smart home gadgets). This was formalized in a decision (Decision 736/QD-BTTTT) that lists mandatory security features for IoT products in Vietnam. These requirements include having unique default credentials or forcing password changes on setup, providing secure update mechanisms for device firmware, implementing data protection measures, and ensuring devices undergo security testing. The regulation also often requires manufacturers to have a point of contact for vulnerability reporting. Vietnam's move mirrors the global trend initiated by standards like ETSI EN 303 645 but makes it enforceable within the country's jurisdiction. Manufacturers of IoT devices in Vietnam (or exporters to Vietnam) need to comply with these rules to sell their products legally. The aim is to reduce the prevalence of easily exploitable IoT devices (which could be hijacked for botnets or spying) on the Vietnamese market, thereby enhancing overall cyber hygiene and protecting consumers.

8. Other

Cyber Resilience Act Compliance Guide for Open Source

Metadata

URL: https://code.inno3.eu/ouvert/guide-cra/-/raw/main/CNLL_inno3_Guide-CRA_VE_1.0.pdf

Publisher: inno³ and CNLL

License: CC-by-SA 4.0

Type:

Publication date: 2024

This guide, put together by European open source advocacy groups, interprets the EU Cyber Resilience Act (CRA) through the lens of open source software development. The CRA will impose certain security and compliance requirements on “products with digital elements.” Many open source maintainers are concerned about how to meet these obligations. The guide likely breaks down the CRA’s key requirements – such as providing security support and updates for a product, having a vulnerability disclosure process, ensuring the software is developed following secure practices, and including technical documentation – and offers suggestions for open source projects to address them. It might recommend things like adopting an open source security best practices badge (to show development follows guidelines), publishing an SBOM for releases, clearly stating how users will be alerted to issues or updates, and possibly working with downstream distributors who can help fulfill CRA duties. It also may clarify which open source scenarios are out of scope of the CRA (for example, maybe software provided without commercial intent). In short, this compliance guide serves as a bridge to help the open source community understand and prepare for the new regulatory landscape

under the CRA, aiming to ensure that open source can remain sustainable and trusted under these rules.

Cyber Resilience Act Requirements Standards Mapping - Joint Research Centre & ENISA Joint Analysis

Metadata

URL: https://www.enisa.europa.eu/publications/cyber-resilience-act-requirements-standards-mapping

Publisher: ENISA

License:

Type:

Publication date:

This joint analysis report by the European Commission's Joint Research Centre (JRC) and ENISA examines how existing cybersecurity standards relate to the requirements of the proposed Cyber Resilience Act (CRA). The CRA will impose certain security requirements on hardware and software products in the EU, and this report maps each specific CRA requirement (for example, on secure design, vulnerability handling, encryption, etc.) to one or more international or European standards that address that topic. The analysis identifies where suitable standards already exist (and could be used to comply with or demonstrate conformity to that CRA requirement) and where there are gaps (i.e. no current standard fully covers a particular requirement, indicating a need for further standardization). This mapping is valuable for industry and regulators: manufacturers can use it to figure out which standards or certifications might help them meet CRA obligations, and standards organizations can see where new work might be needed. Overall, the report serves as a bridge between the high-level legal requirements of the CRA and the practical technical standards that can fulfill them.

Acknowledgments

The following people have contributed to this document either directly or indirectly (e.g. by raising questions):

Anthony Harrison, Ayan Sinha Mahapatra, Christopher "CRob" Robinson, Dick Brooks, Dirk-Willem van Gulik, Jakub Zelenka, Jan Lübbe, Jordan Maris, Lars Francke, Luis Villa, Maarten Aertsen, Marta Rybczynska, Maxim Baele, Merlijn Sebrechts, Pierre Pronchery, Roberto Di Cosmo, Roman Zhukov, Timo Perala, and Tobie Langel.

If you have contributed to this document and aren't properly acknowledged or if you want to edit or remove your name, please let us know by opening an issue and we will fix this right away.

This document was compiled, curated, and edited by Maxim Baele (Toreon/OWASP) and Tobie Langel (UnlockOpen/Eclipse Foundation).

Annex I - LLM Usage

Prompt used:

I linked to a spreadsheet on google drive. These are resources that are relevant for organizations that need to adhere to the EU cyber resilience act. Add a new column "E" in sheet 1 and insert a brief summary of the resource (1 paragraph). Do this in a dry, neutral and academic style, resembling a wikipdia summary. all types of resources are to be taken at face value.

the url is the authorative resource. and yes the summary should reflect the content

LLM used:

OpenAI ChatGPT 4.5 using "deep research".