SECURE
Cyber Resilience for SMEs

The CRA's Essential Cybersecurity Requirements: Annex I, Part I

20/10/2025

Funded by the European Union

EU Funding Statement: Funded by the European Union under GA No 101190325. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Cybersecurity Industrial, Technology and Research Competence Centre.
Neither the European Union nor the granting authority can be held responsible for them.

ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

ECCC disclaimer: The project is supported by the European Cybersecurity Competence Center and its members

DISCLAIMER

This document is part of Deliverable D4.1 'Guidelines and Materials for SMEs CRA Compliance' of the SECURE Project.

First author: *Centre for Cybersecurity Belgium (CCB)*

Second author: *Autoritatea Pentru Digitalizarea Romaniei-NCC-RO (ADR-NCC-RO)*

Table of contents

List of Tables and Figures

Abbreviations

API – Application Programming Interface

APT – Advanced Persistent Threats

BSIMM – Building Security in Maturity Model

CI/CD – Continuous Integration and Continuous Delivery/Deployment

CRA – Cyber Resilience Act

CVD – Coordinated Vulnerability Disclosure

CVE – Common Vulnerabilities and Exposures

CVSS – Common Vulnerability Scoring System

DAST – Dynamic Application Security Testing

DoS – Denial of Service

DREAD – Damage, Reproducibility, Exploitability, Affected Users, and Discoverability

ENISA – European Union Agency for Cybersecurity

EOL – End of Life

EPSS - Exploit Prediction Scoring System

ETSI – European Technical Standard Institute

EU – European Union

FIRST VCMM - FIRST Vulnerability Coordination Maturity Model

GDPR – General Data Protection Regulation

HTTPS/TLS – Hyper Text Protocol Secure/Transport Layer Security

ICS – Industrial Control System

IEC – International Electrotechnical Commission

IoT – Internet of Things

IPSec – Internet Protocol Security

ISO – International Standards Organisation

JTAG – Joint Test Action Group

LINDDUN – Linking, Identifying, Non-repudiation, Detecting, Data Disclosure, Unawareness and Non-compliance

MFA – Multi-Factor Authentication

MITRE ATT&CK – Adversarial Tactics, Techniques and Common Knowledge

MQTT – Message Queuing Telemetry Transport

NIST – National Institute of Standards and Technology (United States)

NIST SSDF – NIST Secure Software Development Framework

NTIA – National Telecommunications and Information Administration (United States)

OPENSSF VEX – Open-Source Security Foundation Vulnerability Exploitability eXchange

OS – Operating System

OTA – Over The Air

OWASP – Open Worldwide Application Security Project

OWASP ASVS – OWASP Application Security Verification Standard

OWASP SAMM – OWASP Software Assurance Maturity Model

PDE – Product with Digital Elements

PSIRT – Product Security Incident Response Team

SAST – Static Application Security Testing

SBOM – Software Bill of Materials

SCA – Software Composition Analysis

SIEM – Security Information and Event Management

SME – Small and Medium Enterprise

SOC – Security Operations Centre

SQL – Structured Query Language

SSDLC – Secure Software Development Lifecycle

SSL – Secure Sockets Layer

STRIDE – Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege

TPM – Trusted Platform Module

UART – Universal Asynchronous Receiver-Transmitter

USB – Universal Serial Bus

VPN – Virtual Private Network

Introduction

In order to comply with the **Cyber Resilience Act** (CRA), Regulation (EU) 2024/2847, as manufacturer, the CRA stipulates a multitude of requirements and obligations. Key within these is that you must ensure that the product with digital elements (PDE) you place on the market "has been designed, developed and produced in accordance with the essential cybersecurity requirements set out in Part I of Annex I"[1]. Annex I contains two parts – Part I focuses on cybersecurity requirements relating to the product's properties, and Part II considers vulnerability handling requirements. Part I further consists of two points, the first of which stipulates that

> *"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"*[2].

Point 2 specifies the requirements your products must comply with.

This guideline, developed within the **SECURE Project**[3] with the aim to **support Small- and Medium Enterprises (SMEs)**, delves into both of the Annex I, Part I points, providing **non-exhaustive practical and technical suggestions, examples, and approaches to support your compliance with each requirement**. It is important to note that any reference made to existing standards, tools, and frameworks is of a purely **suggestive nature** with the intention to make the CRA requirements as tangible as possible. Recommendations made are based on recognised best practices and common approaches. Both the tools referred to within the guideline and the guideline itself will be updated as the development of specific CRA standards and European Commission implementing measures further progresses throughout the adaption period from 2024 to 2027.

[1] Art. 13(1), CRA.

[2] Annex I, Part I(1), CRA.

[3] The 'Strengthening EU SMEs Cyber Resilience' (SECURE) project offers financial support and guidance for SMEs to comply with the CRA.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

The CRA's Essential Cybersecurity Requirements: Annex I, Part I

1. Risk-Based Cybersecurity Approach

Point 1 of Annex I, Part I stipulates that

> *"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**"*[4].

This point is central to the CRA and is grounded in the principle of "security by design and by default". In essence, it means that you must develop a **risk-based cybersecurity approach** for your product. This risk-based tailoring should be defensible, documented, and proportionate. Think of CRA compliance as a "traceable journey":

product context → risk → controls → proof

This journey's rationale should be diligently documented within the mandated technical documentation[5], crucial for compliance and auditability.

In practice, manufacturers must:

1. Assess cybersecurity risks associated with the PDE throughout its lifecycle;
2. Tailor security measures to the level of risk (e.g., a smart thermostat vs. an industrial control system);
3. Consider threat models, attack surfaces, and potential impact on users and systems.

A crucial first step in your compliance with the CRA is thus to conduct a **Risk Assessment** for your PDE. This chapter delves into how to conduct such a risk assessment by setting out possible approaches and offering technical suggestions.

Before diving into the details, a two-page **Risk Assessment 101** overview is provided below to refresh your memory. The elements resurface in greater detail within the first chapter of this guideline, which you are advised to consult. However, for accessibility purposes, the simplified summary sets out how risk assessments are generally approached[6].

[4] Annex I, Part I(1), CRA.

[5] Art. 31, CRA.

[6] It is crucial to note that official guidance on how to conduct the risk assessment for the CRA specifically is still to be developed by the European Commission. The guidance provided here summarises the main steps of any risk assessment approach. Any other approach, standard, or methodology can be used as long as it is in line with the reported approach.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Risk Assessment 101

When conducting a risk assessment, **six steps** can be considered:

1) Identification of **assets** and **threats** = *identify each asset (what needs protecting) according to its exposure to a threat (what could go wrong);*
2) Assessment of **vulnerabilities** = *evaluate the vulnerabilities*;
3) Consideration and evaluation of **impacts** and **likelihood** = *plot the impacts and likelihood of vulnerabilities → this results in specific 'risks'*;
4) Risk **analysis** and **acceptance** = *plot each risk and consider your acceptance level to prioritise your actions;*

Additional to this risk assessment, two final steps include:

5) Implementation of **mitigation measures** = *select and apply security controls for each risk;*
6) Monitoring and reassessment = *monitor the threats and risks throughout each stage of the PDEs lifecycle and keep the risk assessment up to date.*

For **steps one to three**, you can develop a **matrix** that allows you to classify each threat, vulnerability, and impact as corresponding with a certain risk level and score. To do so, you must first define what each level (and score) means for you through descriptive tables[7].

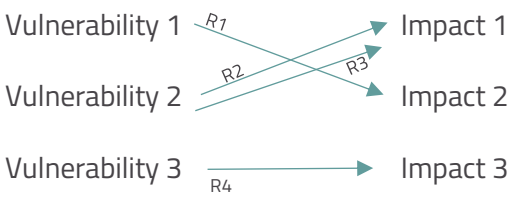
- Low
- Low-Medium
- Medium
- Medium-High
- High

Table 1:
Example Matrix

Risk/Threat	Threat 1	Threat 2	Score
High			10
Med High			
Medium			
Low Med			
Low			0

Linking vulnerabilities to impacts subsequently allows you to visualise the different risks (R):

Figure 1:
Visualisation of Risks

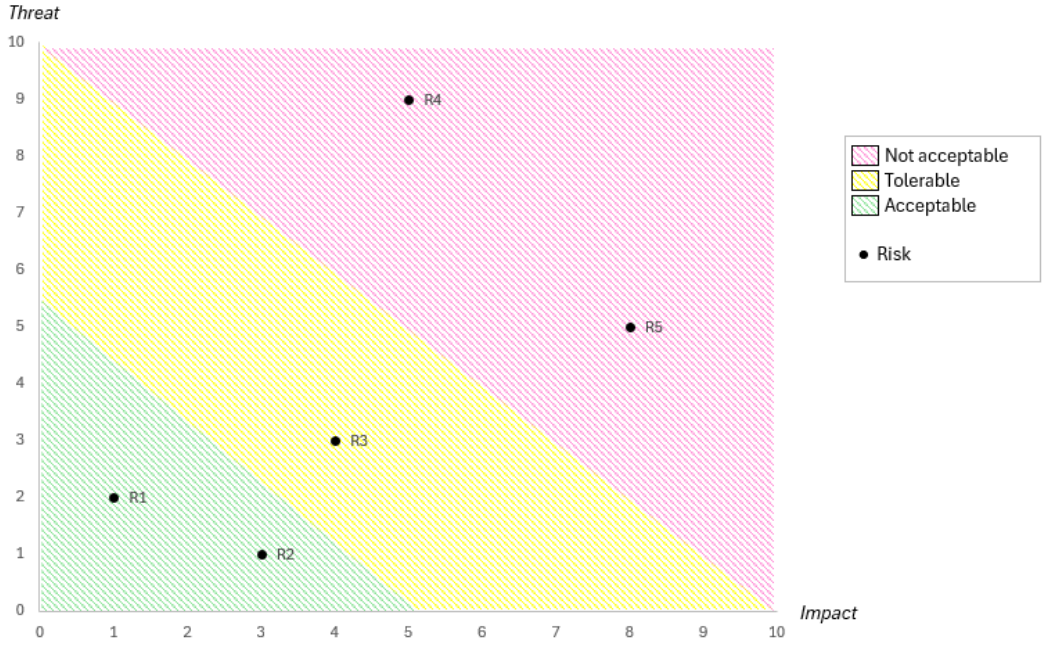


[7] For example, for threat occurrence, a 'Low' level could mean a threat that can strike once every 10 years; whereas 'High' could mean a threat that can strike once every week. You need separate descriptive tables for threats, vulnerabilities and impacts, and risks - the latter supports the definition of your acceptance level.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

For **step four**, it is important to plot the encountered risks and define your level of acceptance to these risks – this is often done through colour coding, for example:

Figure 2:
Risk Acceptance Plot



This allows you to prioritise where to take action and develop mitigation measures and security controls for each risk to reduce the residual risks to an acceptable level.

As stated earlier, these elements of the risk assessment are dealt with in greater detail below (chapter 1 of this guideline) by providing examples and recommended tools and frameworks to support you.

1.1. Lifecycle Cybersecurity Risk Assessment

As the cybersecurity risks associated with your PDE must be assessed throughout the product's lifecycle, conducting a **Lifecycle Cybersecurity Risk Assessment** entails **identifying, analysing, and mitigating cybersecurity risks** at every stage of a product's lifecycle:

1. Design
2. Development
3. Production
4. Deployment
5. Operation and Maintenance

6. End-of-Life (EOL)

The risk assessment must be continuously updated throughout the 'support period'[8], a period of at least five years (or if the product's lifetime is shorter than five years, at least until the end of the product's lifetime).

1.1.1. Key Steps in Lifecycle Risk Assessment

When conducting the risk assessment, six steps can be considered, each of which is clarified in Table 2 below.

Table 2:
Key Steps in Lifecycle Risk Assessment

Key Step	Clarification and Suggestions
1. Identify Assets and Threats	Distinguish between: • Assets: what needs protection? e.g., firmware, user data, communication channels. • Threats: what could go wrong? e.g., malware injection, unauthorised access.
2. Analyse Vulnerabilities	Use tools, such as: • Static code analysis; • Software composition analysis (SCA); • Penetration testing; • Threat modelling[9] (e.g., STRIDE, DREAD). Note: CVSS and STRIDE can be used together in a threat modelling process. STRIDE can help identify potential threats, and CVSS can then be used to assess the severity

[8] Art. 13(8), CRA: The support period is to be determined by the manufacturer, taking into consideration the time during which the product is expected to be in use, user expectations, the product's nature (purpose), and Union law.

[9] Threat modelling is further discussed under point 1.3.1

3. Evaluate Risk Impact and Likelihood	of vulnerabilities related to those threats, allowing for better prioritisation of mitigation efforts[10]. Use a risk matrix to prioritise based on: • Impact (e.g., data breach, system failure); • Likelihood (e.g., known exploit, attack surface).
4. Analyse the Risks and their Acceptability	Define your acceptance level and plot your risks according to the threat and impact to classify and prioritise your actions.
5. Implement Mitigations	Apply security controls: e.g., encryption, authentication, secure boot.
6. Monitor and Reassess	Continuously monitor for new threats and update risk assessments accordingly.

1.1.2. Use Cases by Lifecycle Stage

To make the Lifecycle Risk Assessment more tangible, Table 3 below offers an overview of use cases, including an example risk and a mitigation strategy, per lifecycle stage.

Table 3:
Use Cases by Lifecycle Stage

Lifecycle Stage	Use Case	Risk	Mitigation
Design	Smart Home Camera	Unauthorised video feed access	Implement end-to-end encryption and secure default settings
Development	Device Firmware	Buffer overflow vulnerability	Use secure coding practices and automated vulnerability scanning
Production	Industrial IoT Gateway	Compromise during manufacturing	Secure supply chain and hardware root of trust,

[10] CVSS rates vulnerability severity while risk also factors business impact and likelihood.

			tamper-evident seals and secure provisioning
Deployment	Consumer Router	Default credentials left unchanged	Force password change on first use
Operation and Maintenance	Connected Vehicle	Unpatched software vulnerabilities	OTA (Over-The-Air) updates with integrity checks
End-of-Life	Smart Thermostat	Abandoned device with exploitable firmware	Provide secure decommissioning instructions and data wipe, security update policy at EOL and data export for users

1.1.3. Tools and Frameworks to Support You

The list below highlights several tools and frameworks that may provide support, although a more exhaustive list and harmonised standards are still under development.

- ISO/IEC 27005 – Risk management
- NIST SP 800-30 – Risk assessment methodology
- ENISA Threat Landscape – Updated threat intelligence
- OWASP ASVS – Application security verification
- STRIDE model
- DREAD risk assessment model
- LINDDUN
- CVSS

1.2. Tailored Security Measures

The second dimension to a risk-based cybersecurity approach is to tailor security measures to the level of risk. Tailoring must be proportionate to the risks identified per Annex I, Part I(1). Let's unpack what this means with six clear steps, principles, and two example product cases: a smart thermostat (low-to-moderate risk) and an industrial control system (ICS) (high risk).

1.2.1. Perform Risk Classification of the Product

When performing a product-specific risk assessment, it is important to consider the following dimensions.

- **Threat exposure**: Is the product
  - Internet-connected?
  - Widely deployed?
  - Public-facing?
  - Intended use vs. reasonably foreseeable misuse?
- **Impact of compromise**: What would the impact be on safety, financial loss, privacy, critical infrastructure?
- **Attack attractiveness:** Would it be a steppingstone for lateral movement?
- **User profile:** Consumer, SME, critical infrastructure operator?

Based on these considerations, the product can be classified as Low risk- acceptable, Moderate risk- tolerable, or High risk - unacceptable[11].

1.2.2. Define Security Objectives per Risk Level

Table 4 illustrates how the security objectives can then be tailored to the previously-established product risk level.

Table 4:
Security Objectives per Risk Level

Risk Level	Security Objectives
Low (e.g., smart thermostat)	• Prevent trivial exploitation; • Ensure privacy; • Maintain update capability.
Moderate	• Detect and mitigate known attack vectors; • Enforce authentication;

[11] This requires you to define in advance how to score these elements using a matrix, and how the scores correspond with risk levels. For a more precise classification, five, instead of three, risk levels can be considered: Low, Medium-Low, Medium, Medium-High, High risk.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

	• Secure communication.
High (e.g., ICS)	• Hardened security posture; • Defence-in-depth; • Supply chain trust; • Secure boot; • Incident monitoring.

1.2.3. Map CRA Essential Requirements to the Risk Level

Depending on the risk level, the CRA essential requirements may have different practical applications. Table 5 illustrates this for low- and high risk levels and the respective example product cases, smart thermostat and ICS.

Table 5:
CRA Essential Requirements by Risk Level

CRA Requirement	Low Risk	High Risk
Secure-by-Design and Default	Disable debug ports; strong defaults	Full Software Bill of Materials (SBOM[12]); secure boot; hardened OS
Vulnerability Handling	Public CVD policy; security.txt; monitor inbox; patch mechanism	Coordinated disclosure; PSIRT; rapid response
Logging and Monitoring	Local event logs	Remote logging; SIEM integration
Access Control	Pin or app authentication	Role-based access; MFA; least privilege
Update Mechanism	OTA updates with user consent	Signed updates; fail-safe rollback

[12] SBOM further clarified under section 4.1.

Protection from Unauthorised Access	Basic firewall rules	Host intrusion detection systems; firmware integrity checks

1.2.4. Use Threat Modelling to Refine Measures

For threat modelling, STRIDE, LINDDUN or attack trees can be applied to validate adequacy of controls.

For the example product cases, this means:

- **Smart thermostat**: focus on Spoofing, Tampering, and Denial of Service;
- **ICS**: cover all STRIDE threats and advanced persistent threats (APTs).

1.2.5. Select Controls per Risk Level

Table 6 suggests different controls by security domain for the practical example cases, smart thermostat (low risk) and ICS (high risk).

Table 6:
Controls per Risk Level

Security Domain	Low Risk	High Risk
Authentication	App-based authentication; default password change	MFA; certificate-based access control
Firmware Security	Signed firmware; OTA updates; fail-safe rollback	Secure boot; TPM integration; supply chain assurance
Communication	HTTPS/TLS	VPNs; IPSec; network segmentation; zero trust
Monitoring	Basic log rotation	Real-time logging; anomaly detection; SOC integration; time-synchronised logs
User Interface	Simple settings panel	Fine-grained admin console; audit trail capabilities

1.2.6. Document Compliance Evidence

As stated earlier, a key element to compliance with the CRA, is documenting the following evidence, at minimum:

- Risk classification rationale;
- Control decisions linked to risks;
- Testing and validation results;
- Update and vulnerability handling policies;
- Secure development lifecycle alignment (e.g., ISO/IEC 27034, IEC 62443-4-1);
- Traceability matrix mapping risks → controls → verification tests → evidence (to be kept in Technical Documentation)

1.2.7. Examples

Table 7 provides additional examples of implementation measures for the low- and high risk practical example cases.

Table 7:
Implementation Measures per Risk Level

Product	Smart Thermostat	ICS
Risk Considerations	• Internet-connected, controls heating in private home; • Privacy-sensitive, but low safety or economic impact.	• Used in critical infrastructure (e.g., water treatment); • High safety and operational impact.
Risk Classification	Low Risk	High Risk

Implementation Measures	• Change default password on first use; • HTTPS communication with backend; • Signed firmware updates; • Local logging only[13]; • Basic vulnerability contact form.	• Secure-by-design hardware with TPM; • Secure boot and signed updates with rollback; • Role-based access and MFA; • Network segmentation and firewall rules; • Logging to central SIEM; • Full SBOM with every update; • Coordinated vulnerability disclosure (CVD) process;

1.3. Considering threat models, attack surfaces, and potential impact on users and systems

The CRA emphasises a risk-based approach to cybersecurity. As shown above, this means that manufacturers should tailor their security measures to realistic threats, points of exposure and potential consequences for users and systems. This is not a hollow directive, but a call for a substantiated, context-aware approach. To effectively implement this obligation, three key concepts must be considered together: threat models, attack surfaces and impact analysis, forming the third and final dimension of the risk-based cybersecurity approach. Let's examine these elements one by one and see how they fit together.

1.3.1. Threat Models: Who attacks, Why, and How?

Threat modelling is a structured process in which you identify who could attack your product, how they would do so and what their motivation is. Think of script kiddies, organised cybercriminals or even state actors. Their motives vary from financial gain to sabotage or espionage, and their skills range from basic to advanced.

[13] To note: local-only logs reduce forensic value, optional export on user consent is recommended.

To structure this, you can use methods such as:

- STRIDE;
- MITRE ATT&CK for known attack techniques;
- LINDDUN for privacy-oriented threats;
- Attack trees or cyber kill chains to map attack paths.

Returning to the practical example cases, this means:

- **Smart thermostat**: threats are often limited to curious neighbours or random attacks, where someone could manipulate the temperature settings or energy consumption;
- **ICS** (e.g., in a water treatment plant): threats are fundamentally different - e.g., APT groups or ransomware gangs try to sabotage physical processes or shut down a business.

The result of threat modelling is a clear list of security objectives specific to the product and its environment.

1.3.2 Attack Surfaces: Where can an attacker get in?

An attack surface is the totality of all points at which an attacker can interact with or influence the system. The more interfaces and access points, the greater the risk.

Typical attack surfaces are:

- Network interfaces such as Wi-Fi, Bluetooth, MQTT or HTTP;
- Local interfaces such as USB, UART, JTAG (for debugging);
- Update mechanisms such as OTA or USB updates;
- APIs, mobile apps, cloud dashboards:
- External components from the supply chain.

Analysing these surfaces involves checking which components are unnecessarily exposed, which services are enabled but not needed, and whether access is adequately protected. Ideally, you should limit the attack surface by:

- Security principles such as minimal exposure, secure defaults and hardening;
- Disabling unused ports or services;
- Authentication and encryption at every interface.

Applied to the examples, this means:

- **Smart thermostat**: will typically use Wi-Fi and possibly Bluetooth, with a simple cloud connection - Debug interfaces may be open during testing and must be deactivated in production;
- **ICS gateway**: will be physically protected, with shielded USB updates, segmented networks and no external interfaces.

Careful mapping of the attack surface is therefore essential to know where security is really needed.

1.3.3 Impact Analysis: What happens if things go wrong?

The final step is to determine the potential impact of a successful attack. The CRA requires security measures to be proportionate to this impact. This includes not only technical damage, but also:

- Hazards for the user (e.g., injury due to temperature control);
- Violation of privacy (e.g., inferring living patterns from thermostat data);
- Loss of availability or business continuity (e.g., factory shutdown);
- Legal liability (e.g., violation of the CRA or GDPR);
- Reputational damage and market risk.

Impact must be considered across multiple dimensions:

- User: from minor inconvenience to life-threatening situations;
- Organisation: from increased helpdesk workload to business interruption;
- Society: from innocent bugs to threats to critical infrastructure.

Here too, proportionality is key - a toy robot does not require the same level of security as a medical pump.

1.3.4 Integration: From Analysis to Measures

When these three building blocks come together — threat model, attack surface and impact — a solid foundation is created for tailoring security measures.

A typical approach looks like this:

1. Define the product use and context;
2. Perform threat modelling to understand actors, motives and attack paths;

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

3. Map the attack surface and identify vulnerabilities;
4. Analyse the impact on users, organisations and society;
5. Select measures based on risk (risk = probability × impact);
6. Document everything for CRA compliance and audits.

For the practical example cases, this means:

- **Smart thermostat**: gets encryption, strong password policy, signed OTA updates, and a simple privacy statement.
- **ICS gateway**: gets secure boot, hardware root of trust, segmented networks, SIEM logging, role management, and a complete SBOM with vulnerability monitoring.

2. Secure Design and Development

Returning to point 1 of Annex I, Part I of the CRA, the risk-based cybersecurity approach and tailoring is grounded in the principle of "security by design and by default", i.e. PDEs must be designed and developed to be secure from the ground up. It is no longer sufficient to add security as an optional layer after the fact; it must be an essential part of the entire product development process. 'Secure by Design', 'Secure by Default' and the use of secure development practices form the core of a resilient digital product strategy. They ensure that security is not an afterthought, but a structurally and demonstrably integrated part of the product – exactly what the CRA requires.

Using international standards such as IEC 62443, ISO 27034, OWASP and ENISA guidelines, manufacturers can efficiently apply these principles while meeting their compliance obligations.

Products must be:

1. **Secure by design**: Security is integrated from the earliest stages of development;
2. **Secure by default**: Default settings must prioritise security (e.g., strong passwords, minimal open ports, etc.)
3. **Securely developed**: Using secure coding practices and threat modelling.

2.1. Security by Design – Secure from the Initial Design

'Secure by design' means that cybersecurity is taken into account from the concept phase in decisions about architecture, component selection and interaction between subsystems. Security must be as fundamental as functionality or user-friendliness.

Practical example:

When designing a smart door lock module, the following decisions are made immediately:

- Apply end-to-end encryption between the app and the lock;
- Store keys securely in a TPM or Secure Element;
- Physically disable debug ports after production.

Relevant standards and guidelines in this regard include:

- IEC 62443-4-1: Requires security integration in the software lifecycle;
- ISO/IEC 27034: Application security in the software development lifecycle;
- NIST SP 800-218 SSDF;
- ENISA Secure Software Development Good Practices.

2.2. Security by Default – Secure without User Configuration

'Secure by default' means that products are delivered with the most secure configuration as standard. The user should not have to guess whether security is enabled. Security is the baseline, not an optional 'advanced setting'.

Examples of secure default settings:

- No shared defaults, enforce first-boot credential setup or passwordless pairing with secure factors;
- Only necessary network ports open (principle of minimal exposure);
- Firmware updates signed and verified by default;
- Logging and audit trail enabled by default for critical functions.

Relevant guidelines in this regard include:

- OWASP Secure Configuration: Best practices for secure default settings;
- NIST SP 800-128: Guide for Security-Focused Configuration Management.

2.3. Secure Coding Practices

The CRA requires that software development be carried out in accordance with proven secure development practices and with continuous attention to threats. This means, among other things:

Secure coding:

- Input validation (against SQL injection, buffer overflows, etc.);
- Use of secure libraries and encryption;
- Fuzz testing and static code analysis.

Threat modelling:
For each component of the software, the following must be assessed:

- Who could attack this?
- How could they do it?
- What would be the impact?

Frameworks such as STRIDE (Microsoft), OWASP Threat Dragon, and MITRE ATT&CK can help to systematically identify vulnerabilities and attack paths.

Relevant standards and guidelines in this regard include:

- OWASP Secure Coding Practices Checklist;
- ISO/IEC 27001 Annex A.14: Security requirements in development;
- ENISA Threat Modelling Guidelines (2022);
- BSI TR-03161 (Germany): Development of Secure Software.

2.4. In concrete terms for Manufacturers

To summarise, an organisation that wants to develop CRA-compliant PDEs should:

- Adopt a secure software development lifecycle (SSDLC), as described in IEC 62443-4-1 or NIST SP 800-218 SSDF;
- Have a code review and testing policy that focuses on vulnerabilities (SAST, DAST, fuzzing);
- Systematically apply threat modelling to every important component;
- Deliver products with standard closed ports, logging enabled and secure access ports;

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

- Adopt a PSIRT function with clear roles and on-call processes;
- Define security quality gates in CI/CD (SAST, DAST, SCA, secrets scan) with fail-the-build policies.

3. Lifecycle Security Management

Beyond the secure design, development and production of PDEs, your product also needs to remain secure throughout its entire lifecycle. Digital products evolve — and so must their security. This means that security must be continuously managed and considered also after the PDE has been brought onto the market.

Concretely, manufacturers are required to[14]:

1. Monitor vulnerabilities continuously;
2. Provide timely security updates and patches;
3. Maintain a vulnerability disclosure policy and communicate risks transparently to users and regulators.

3.1. Continuous Vulnerability Monitoring

Once a product is on the market, manufacturers must actively and systematically continue to detect vulnerabilities. This includes:

- Monitoring vulnerability databases such as the European vulnerability database[15];
- Monitoring supplier advisories;
- Tracking Common Vulnerabilities and Exposures (CVEs) relating to components or libraries used;
- Use of SBOM to identify and track dependencies;
- Internal monitoring for new vulnerabilities through bug bounty programs, penetration tests or security audits.

[14] Although vulnerability handling requirements are covered in-depth in Annex I, Part II, they stem from Point 1 and 2 of Annex I, Part I, and are therefore already touched upon in this guideline.
[15] Art. 17(5), CRA.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Example:

A manufacturer of network cameras uses open-source firmware modules. The CVE database reveals that one of these modules contains a critical vulnerability (e.g. CVE-2023-XXXXX). The manufacturer is obliged to monitor and evaluate this information and, if relevant, take appropriate action.

Relevant sources in this regard include:

- CVE;
- EPSS (Exploit Prediction Scoring System);
- ENISA Vulnerability Management Guidelines;
- ISO/IEC 30111: Vulnerability handling processes.

3.2. Timely Security Updates and Patches

The CRA requires manufacturers to respond quickly to known vulnerabilities and to distribute security updates free of charge and effectively throughout the support period.

These updates must:

- Be digitally signed and validated;
- Have a fail-safe rollback mechanism;
- Be automatically installable with opt-out options and/or with minimal user interaction;
- Remain available for at least 10 years after release, or for the remainder of the support period (whichever is longer).

Example:

A manufacturer of smart thermostats discovers a vulnerability in the Wi-Fi stack. Within two weeks, a security patch is developed, tested and distributed via a signed OTA update. Users receive a clear notification and the update is automatically installed when the device is restarted.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Relevant sources in this regard include:

- ISO/IEC 29147: Coordinated vulnerability disclosure;
- NIST SP 800-40: Guide to Enterprise Patch Management;
- ETSI EN 303 645: Security baseline for consumer IoT (including software update mechanisms).

3.3. Vulnerability Reporting and Transparent Communication Policy

Transparency is essential. The CRA requires manufacturers to:

- Publish a CVD policy;
- Provide a contact point (e.g., security@company.eu) for reports;
- Inform users and authorities such as ENISA or the national supervisory authority quickly in the event of serious risks;
- Communicate transparently about available patches, mitigations and remaining risks.

Example:

An ethical hacker reports a critical vulnerability in a connected alarm system via the manufacturer's public CVD platform. Within 72 hours, receipt is confirmed and, after internal analysis, ENISA is informed via the ENISA single reporting platform (national endpoints). A patch is rolled out within three weeks and all users are notified of the risk and the solution via email and app notifications.

Relevant sources in this regard include:

- FIRST Vulnerability Coordination Maturity Model (VCMM);
- ISO/IEC 29147: Guidelines for vulnerability disclosure;
- ENISA Coordinated Vulnerability Disclosure Guidelines (2022);
- OpenSSF VEX (Vulnerability Exploitability eXchange).

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

4. Supply Chain Security

The CRA recognises that a product is never completely 'independent': it consists of dozens, sometimes hundreds, of components from external suppliers, open-source projects and hardware partners. That is why the CRA sets explicit requirements for managing cybersecurity risks within the supply chain.

In practice, manufacturers must:

1. Maintain an up-to-date and transparent overview of the software components used via a SBOM;
2. Require suppliers to comply with CRA-compliant security;
3. Actively monitor and manage risks associated with open-source and external dependencies.

4.1. Software Bill of Materials (SBOM)

A SBOM is similar to an ingredient list for software: it contains an overview of all components, versions and origins of software elements used, including open-source libraries.

The CRA requires manufacturers to maintain a SBOM and be able to submit it to regulators and authorities upon request. SBOM publication for users is optional[16]. This SBOM forms the basis for:

- Vulnerability analysis (e.g. via CVE tracking);
- Impact assessment in the event of zero-days;
- Supply chain audits.

Example:

A manufacturer of smart routers draws up an SBOM that clearly states that the product uses:

- OpenSSL 1.1.1n;
- BusyBox 1.35.0;
- A modified version of an open-source firewall module.

[16] If made available for users, clarify where/how users can access it.

When a vulnerability in OpenSSL (CVE-2022-XXXX) is disclosed, the manufacturer can immediately check whether the product is affected and respond appropriately.

Relevant sources in this regard include:

- CycloneDX, SPDX: SBOM formats (also recommended by ENISA and NTIA);
- ISO/IEC 5230 (OpenChain): Supply chain software compliance;
- OpenSSF tooling for SBOM generation and vulnerability detection.

4.2. Security Requirements for Suppliers

The CRA also requires manufacturers to ensure that their suppliers and external developers adhere to security requirements comparable to those of their own team. Responsibility cannot be passed on; vulnerabilities in third-party components can also lead to CRA compliance obligations.

In concrete terms, this means:

- Inclusion of cybersecurity clauses in supplier contracts;
- Performing security due diligence when selecting software suppliers;
- Periodically verifying that partners comply with, for example:
    - ISO/IEC 27001 (information security);
    - IEC 62443-4-1 (secure product development);
    - OWASP SAMM or BSIMM maturity models.
- Contractual audit rights and minimum assurance levels (e.g., certification of conformity claims) for critical components;
- Notification within 24h of supplier-discovered critical vulnerabilities affecting your PDEs.

Example:

A manufacturer of medical IoT devices works with a software supplier in Asia. The contract stipulates that this supplier:

- Develops securely in accordance with IEC 62443-4-1;
- Document all open-source components used;
- Maintains a vulnerability policy with mandatory reporting within 24 hours.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

4.3. Risk Management of Open-Source and External Libraries

Open-source software offers many advantages, but also comes with risks: vulnerabilities, missing updates, unclear licences or unreliable maintainers. The CRA requires manufacturers to actively manage and monitor these risks.

Best practices include:

- Use dependency scanners (e.g. OWASP Dependency-Check, Snyk, Trivy);
- Automatic alerts for vulnerabilities (e.g. via GitHub Advisories);
- Only use maintained and mature open-source projects;
- Apply security gates in CI/CD pipelines (blocking builds with known CVEs);
- Use VEX to reduce noise from non-exploitable CVEs;
- Require minimum maintainer responsiveness when selecting OSS.

Example:

A manufacturer uses a popular JavaScript library (e.g. Log4j) in a web interface. Upon discovering Log4Shell (CVE-2021-44228), the manufacturer knows exactly which versions are affected through SBOM analysis and can segment and patch the affected products.

Relevant sources in this regard include:

- NIST SSDF (Secure Software Development Framework);
- ENISA OSS Security Guidelines;
- OpenSSF Scorecard: Objective quality assessment of open-source projects.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Conclusion

This guideline presents a first **technical translation of the CRA's Annex I, Part I requirements into practical suggestions and recommendations**. It highlights **four components** that are critical to your compliance with the CRA: (1) a risk-based cybersecurity approach, which entails a risk assessment, tailored security measures, and a consideration of threat models, attack surfaces and impacts; (2) the secure-by-design/default principle; (3) security management duties throughout your PDE's lifecycle; (4) supply chain considerations and controls. For each component, **practical suggestions** are made on how to implement and comply with these obligations based on recognised best practices and standards. These are, however, subject to change pending current standards' discussions and further regulatory developments. As next steps for SMEs, it is recommended to consult additional guidelines on the **SECURE repository**, such as the **CRA Methodological Compliance Assessment Framework** for a step-by-step toolkit and checklist on compliance with the CRA beyond Annex I, as well as **CRA 101: Understanding CRA Obligations** for a beginner-friendly condensed overview of your legal obligations under the CRA.

Funded by the European Union
ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE