FOSSID

Software Supply Chain Integrity and SBOM Obligations under the EU Cyber Resilience Act

A Practical Guide for OSPO Leaders, Security Officers, and Product Teams

Executive Summary

The EU Cyber Resilience Act (CRA) marks a significant shift for anyone building or selling software in Europe. It brings the EU in line with global efforts to make software supply chains more transparent and accountable. The CRA embeds security and transparency requirements into every stage of software development and makes failure to meet essential cybersecurity requirements a matter of regulatory non-compliance. For software producers, this changes the rules. SBOMs, vulnerability management, and open source governance are now legal obligations.

This paper explains what the CRA expects, how it connects to existing open source and supply chain practices, and what practical steps engineering and Open Source Program Office (OSPO) leaders should take now to prepare. The paper also covers how to build compliance into existing workflows and how to do it without slowing development. It also offers a CRA Readiness Checklist, a three-page guide to help you assess CRA compliance readiness.

Table of Contents

A New Era of Software Accountability	4
SBOM: The Core of Sofware Supply Chain Transparency	5
Vulnerability Management under the CRA	6
Embedding CRA Compliance into the Software Supply Chain	7
Aligning Teams Around CRA Compliance	8
Practical Steps for OSPO Leaders	8
Common Challenges and Mitigation Strategies	11
CRA Full Compliance	12
Turning Compliance into Competitive Advantage	13
Conclusion	13
Appedix A: CRA Readiness Checklist	14

A New Era of Software Accountability

The Cyber Resilience Act (CRA) marks the EU's effort to make digital products safer by design. It applies to any product with digital elements sold in the EU, from IoT devices to software applications. The regulation distinguishes between default, important and critical products based on their cybersecurity risk.

Many products are expected to fall into the default category and are subject to baseline requirements. Products deemed important are further split into Class I and Class II, depending on their criticality and potential impact. Critical products and important products in Class II require third-party conformity assessments, while default products and important products in Class I may generally rely on internal conformity assessments and controls.

It shifts responsibility from consumers to producers, requiring documented evidence of security controls, ongoing risk management, and transparency into third-party components. For most organizations, the CRA is both a regulatory requirement and a structural change to how software is built, released (placed on the market), and maintained. In practice, this means manufacturers must be able to demonstrate security and compliance across the entire product lifecycle, not just at release.

In a nutshell, the CRA makes software producers legally accountable for the security and transparency of their products. To put the timeline in perspective, here are the key milestones shaping the CRA implementation:

Date	Milestone
December 10, 2024: Entry into force	The CRA entered into force 20 days after publication in the Official Journal (publication 20 Nov 2024). From that date the clock starts on the staged application of different parts of the Regulation.
June 11, 2026: Chapter IV applies	Chapter IV (Articles 35–51) on notification of conformity assessment bodies takes effect from 11 June 2026. Member States must then notify which bodies are authorized to carry out CRA conformity assessments. The Regulation also requires Member States to strive to ensure enough notified bodies by 11 December 2026 to avoid bottlenecks. This is about the readiness of the conformity ecosystem rather than product reporting.

Date	Milestone
September 11, 2026: Reporting obligations (actively exploited vulnerabilities and severe incidents) start	Article 14 (manufacturer reporting obligations) applies from September 11, 2026. From that date, manufacturers must use the ENISA single reporting platform to submit early-warning notices (within 24 hours of awareness), a vulnerability/incident notification (within 72 hours), and follow-up/final reports (timing depends on the notification type). The Article covers actively exploited vulnerabilities and severe incidents.
December 11, 2027: Full application of the Regulation	The majority of CRA obligations apply from December 11, 2027 (36 months after entry into force). From that date, manufacturers must comply with the complete set of requirements (technical documentation, SBOM elements, conformity assessment / CE-type processes where applicable, secure-by-default, update obligations, etc.). Products placed on the market before December 11, 2027, are only brought in if they are substantially modified after that date. Still, Article 14 (reporting obligations) applies to all in-scope products even if placed on the market earlier.

To understand how to comply with the CRA, we need to start with its most impactful requirement: the SBOM.

SBOM: The Core of Software Supply Chain Transparency

The SBOM is the foundation of CRA compliance. It provides a detailed inventory of every component that makes up a software product, including open source dependencies. The goal is simple: know what you ship so you can react fast when new vulnerabilities appear.

Unlike voluntary frameworks such as NIS2 or IEC 62443-4-1, the CRA turns SBOM generation into a legal obligation tied to product conformity. The CRA requires manufacturers to generate and maintain SBOMs as part of their technical documentation to support vulnerability management and risk assessment. These SBOMs must be available to market surveillance authorities upon request. While organizations may choose to share SBOMs with customers under contractual arrangements or transparency initiatives, this is not a general requirement under the CRA. In practice, this means software producers will need an automated, reliable SBOM generation process embedded into their CI/CD pipelines with updates triggered by each release, patch or dependency change.

SBOMs must cover at least the top-level third-party components and dependencies included in a product, including open source software (OSS). These SBOMs should include core metadata such as the component name, version, supplier, and license information. While many organizations choose to extend SBOM coverage to transitive dependencies and enrich them with vulnerability data as a best practice for effective vulnerability management, the CRA does not explicitly mandate full transitive dependency coverage within the SBOM itself.

The EU has not mandated a specific SBOM format, but SPDX and CycloneDX are widely recognized and accepted standards. What matters is accuracy and currency: the SBOM must reflect what is actually shipped. SBOMs should therefore be kept up to date for each release, patch, or change affecting software components.

Under the CRA, SBOMs become a formal part of the product's technical documentation and a regulatory mechanism for demonstrating supply-chain awareness, rather than just an internal inventory. However, visibility alone is not sufficient. Manufacturers are also required to act on this information by monitoring for vulnerabilities, assessing their impact, and remediating and reporting issues in line with the CRA's vulnerability-handling obligations.

Vulnerability Management under the CRA

The CRA doesn't stop at listing components. It demands continuous monitoring and management of vulnerabilities. Producers are required to monitor, assess, and remediate security issues throughout a product's lifetime. When a vulnerability is identified, organizations must disclose it responsibly and patch affected products in a timely manner. They must report any actively exploited vulnerability or severe incident within 24 hours of awareness using ENISA's reporting platform. A robust Product Security Incident Response Team (PSIRT) process will become essential for coordinating vulnerability intake, triage, and disclosure.

This requirement changes how engineering or development teams think about software maintenance. They need structured processes for tracking vulnerabilities, triaging their severity, and reporting them within the CRA's prescribed timeframes, typically within 24 hours for active exploitation cases. The CRA aligns with emerging EU guidance on coordinated vulnerability handling and disclosure, encouraging coordination between developers, security teams, and national authorities.

Organizations need to show that their vulnerability management process is structured, consistent, and trackable.

Three key areas to keep focus on:

- **Active exploitation:** if a vulnerability in your product is being actively exploited, you must report it within 24 hours to ENISA and national Computer Security Incident Response Teams (CSIRTs).
- **Serious security incidents:** if a product suffers or could suffer a serious compromise, you'll need to report details. There are shorter windows for initial alert and more detailed follow-up.
- **Support period obligation:** Manufacturers must support their products with security updates for a specified period, often tied to product lifecycle or market expectations. Manufacturers must make clear what that period is.

To stay responsive, compliance has to be built directly into day-to-day development and release workflows. We will explore this topic in the next section.

Embedding CRA Compliance into the Software Supply Chain

Compliance with the CRA requires operational integration across development, procurement, and release management. Most CRA requirements align with modern DevSecOps practices such as dependency scanning, signed builds, and provenance tracking.

A practical approach begins with mapping your software supply chain: identify where code comes from, how it's assembled or integrated, and where risk points exist. Then, use automation to capture that information continuously. Modern tools for Software Composition Analysis (SCA) can automatically generate SBOMs, detect outdated dependencies, and feed vulnerability data directly into your issue tracking systems.

The next layer is governance, where OSPOs play a key role. They can establish open source policies, ensure license compliance, and create a unified process for external vulnerability reporting. By integrating CRA obligations into the OSPO's scope, organizations can manage open source and security compliance under a single operational framework.

Many of the most complex compliance challenges will come from supply chains, especially open source and third-party components:

- If you integrate OSS into a product with digital elements, you're responsible for tracking it.
- Upstream OSS projects often don't provide full metadata (versioning, dependency graphs, vulnerability info). You may need to fill in the gaps.

This shift only works if teams coordinate and align their efforts. Readiness depends on how well the organization aligns around it.

Aligning Teams Around CRA Compliance

The CRA affects engineers, legal, procurement, IT, and product management. To meet its obligations, these functions must work together under a shared understanding of what "secure by design" means in practice.

The first step is ownership. Every product should have a clearly defined security and compliance owner who ensures SBOM accuracy, vulnerability reporting, and lifecycle maintenance. Without clear accountability, the risk of gaps or delays increases.

The second step is training. Developers, product managers, and compliance officers need to understand how CRA affects their daily work, what to document, what to report, and how to use the new tooling effectively. Some organizations are creating "compliance champions" within engineering teams to bridge this knowledge gap.

Finally, leadership must recognize compliance as part of product quality, not as a bureaucratic overhead. Organizations that treat CRA preparation as a product capability will gain a competitive edge when the enforcement phase begins.

Effective CRA readiness depends on cross-functional alignment and strong ownership structures. Once alignment is achieved, organizations can start building the technical and procedural foundation for long-term compliance.

Practical Steps for OSPO Leaders

Sustainable CRA compliance requires automation, monitoring, and documentation. Manual reporting or one-off SBOM generation won't scale across large product portfolios. Therefore, compliance must be a continuous, traceable process baked into development workflows. This ongoing process starts with your CI/CD pipeline. Every

build should automatically produce an SBOM, sign artifacts, and store provenance metadata. These artifacts form the backbone of your technical documentation and evidence for CRA audits. In parallel, vulnerability intelligence should flow seamlessly between your SCA tools, issue trackers, and PSIRT process. When a vulnerability appears, the system should trigger alerts, assign remediation tasks, and record the timeline of actions taken. This kind of traceability is what regulators will look for.

Documentation is another cornerstone. CRA requires organizations to maintain technical documentation for at least 10 years after a product's last release. This requirement means maintaining consistent, accessible records that show how the organization had handled vulnerabilities and what components were included in each SBOM.

Here's a structured way to operationalize CRA readiness:

1. **Scope your portfolio**:
    - Identify which products with digital elements (PDEs) fall under CRA obligation
    - Map internal owners for each PDE (engineering, legal, security, product)
    - Assign a CRA lead or steering group, ideally anchored in the OSPO or product security function

2. **Conduct a compliance gap analysis:**
    - For each product, assess where you stand on SBOM coverage, vulnerability handling, reporting, and update mechanisms
    - Compare against the CRA requirements
    - Highlight dependencies on third parties or suppliers that could affect compliance
    - Document risk areas (e.g., incomplete SBOM coverage, missing PSIRT process)

3. **Define internal standards:**
    - Pick your SBOM format(s)
    - Choose SCA tools
    - Specify frequency for SBOM updates
    - Versioning rules
    - Component metadata expectations

4. **Build tooling and automation:**
    - Integrate SBOM generation into your build/CI/CD pipelines
    - Use or upgrade SCA (Software Composition Analysis) tools to detect vulnerable dependencies
    - Automate versioning and component tracking
    - Validate SBOM accuracy against SPDX or CycloneDX standards
    - Set up a review process to update SBOMs for every release or patch

5. **Design vulnerability management workflow:**
    - Identify who is responsible (PSIRT, security team, product owner)
    - Establish detection: correlation between SBOM data & vulnerability feeds
    - Define reporting trigger: actively exploited vs potential vulnerabilities
    - Build procedures for reports: internal, then to authorities (within required windows)
    - Implement processes for identifying, triaging, and disclosing vulnerabilities within 24 hours of awareness
    - Document ENISA reporting workflows and contact points

6. **Update supplier contracts:**
    - Require vendors to provide CRA-compliant components, SBOMs, and vulnerability notifications
    - Add clear service-level expectations for response times, remediation, and documentation
    - Establish a supplier audit process

7. **Develop and store technical documentation**
    - Prepare conformity documentation for each PDE, including:
        i. SBOM
        ii. Security design rationale
        iii. Risk assessment
        iv. Compliance evidence (test results, certification)
    - Store documents in an auditable, version-controlled repository

8. **Train teams across functions:**
    - Provide targeted training for developers, product managers, and legal on CRA obligations
    - Create quick-reference guides for vulnerability reporting and SBOM updates
    - Build awareness of deadlines (2026–2027 enforcement milestones)

9. **Run internal audits and dry runs:**
    - Set metrics to track CRA readiness (e.g., SBOM coverage %, time-to-report vulnerabilities)
    - Review progress quarterly with leadership
    - Continuously monitor EU guidance and updates from ENISA or national authorities

Automation and traceability are the pillars of continuous CRA compliance. With systems and processes in place, the final step is to focus on readiness, preparing for audits, and real-world enforcement. In the next section, we will discuss common challenges and how to address them.

Common Challenges and Mitigation Strategies

When the CRA takes full effect, organizations will need to demonstrate compliance through documentation and evidence, including SBOMs, vulnerability reports, test results, and security risk assessments. The most efficient organizations will be those that can produce this evidence at the click of a button.

Audit readiness involves more than storing data; it requires structuring it logically. Ensure that each product's compliance documentation includes a clear lineage from code to component to vulnerability fix. Regulators or conformity assessment bodies will look for traceability and consistency across versions.

Companies that export to the EU should also coordinate with their legal and regulatory teams to understand reporting obligations and potential penalties for non-compliance. Non-conformity could lead to product recalls, fines, or temporary bans from the EU market.

Even with the checklist above, you'll face friction. Here's a list of challenges that usually come up and how OSPOs can plan and mitigate them:

Challenge	Mitigation
Missing or inconsistent component metadata	• Use tools that parse dependency trees • Invest in upstream open source to improve metadata
Transitive dependencies unknown	• Use SCA tools that capture deep dependency graphs • Maintain internal libraries where possible to control versioning
Resource constraints for vulnerability reporting	• Prioritize better-performing products or highest-risk PDEs first • Build shared responsibility among product/security teams
Contractual pushback from suppliers	• Create standard clauses • Show how non-compliance can delay market entry • Make SBOM/data sharing a competitive advantage
Keeping SBOMs up to date post-release	• Automate updates as part of patch releases • Schedule periodic reviews • Set up alerts for upstream vulnerabilities

For organizations that act early, CRA preparation can evolve from a compliance task into a strategic advantage. The earlier you detect and automate these patterns, the easier it becomes to scale CRA compliance across multiple product lines.

CRA Full Compliance

Imagine we're already in December 2027, you're an OSPO leader, and someone asks about compliance. You should be able to show:

- A well-maintained SBOM for each PDE, fully covering dependencies and version history.
- Vulnerability monitoring is now integrated into operations; actively exploited vulnerabilities are reported per CRA's timetable.
- Technical documentation (user instructions, risk analyses, update policies, support periods) is now available.
- Supplier contracts are aligned with the CRA requirements.
- Evidence of compliance audits or internal reviews are documented.

- Internal training is developed and made mandatory, and there is clear process ownership.

Turning Compliance into Competitive Advantage

The CRA might feel burdensome at first, but it can also strengthen your company's position. Customers, partners, and regulators increasingly demand transparency and assurance. By operationalizing SBOMs and vulnerability management, you are achieving two goals: meeting legal requirements and building trust.

Organizations that integrate CRA obligations early can reuse their compliance systems to meet other standards such as ISO/IEC 27001, NIS2 Directive, or US Executive Order 14028. The same visibility that enables compliance also improves software quality, speeds up incident response, and enhances internal confidence in product security. For OSPOs and engineering leaders, the message is clear: treat CRA readiness as a long-term capability. It's an investment in the resilience and credibility of your software supply chain.

The CRA might feel regulatory, but it's really a business differentiator. By making transparency, SBOM accuracy, and vulnerability responsiveness core capabilities, you not only comply, you build trust with customers, regulators, and partners.

Conclusion

The CRA aims to restore confidence in the digital systems that power our economies. For software producers, it's an opportunity to modernize supply chain practices, integrate security into every stage of development, and demonstrate accountability in a tangible way. Organizations that understand the intent of the CRA (transparency, responsibility, and continuous improvement) will adapt smoothly. Organizations that wait until enforcement risk playing catch-up in a world where software integrity is now a legal expectation.

For OSPOs, the CRA is a turning point. Throwing together an SBOM at the last minute won't cut it (not anymore). What will matter is how consistent, transparent, and responsive your supply chain becomes. Integration of SBOMs, transparent vulnerability reporting, and solid documentation, done well, will make your products safer, your brand stronger, and your company more resilient.

The future of compliance is continuous, and it begins with knowing what you ship. You can start today, build layer by layer, measure results, and implement improvements as you do. In 2027, when everyone else is scrambling to check boxes, your team will already be living the requirements. To support you with that building process, we have prepared a CRA Readiness Checklist, a three-page guide to help you assess CRA compliance readiness.

The CRA carries real weight: fines can reach €15 million or 2.5% of global annual turnover, whichever is higher, echoing the GDPR's approach to enforcement. The take-away for organizations is simple: investing early in CRA compliance is prudent and strategic. As Benjamin Franklin reminded us nearly three centuries ago, "An ounce of prevention is worth a pound of cure."

About the Author

Gary Armstrong

Senior Director of Customer Success, FossID

Gary Armstrong is dedicated to empowering businesses to harness the benefits of open source software while ensuring legal compliance and security confidence. Backed by more than a decade of experience delivering open source security and compliance services, Gary shares his insights and best practices through writing and speaking engagements with the open source community.

Appendix A: CRA Readiness Checklist

A practical guide for OSPOs, security, and engineering leaders to assess their CRA compliance readiness. Use this checklist to identify gaps and track progress across governance, SBOM management, vulnerability handling, and documentation readiness.

1. Assign owners for each domain, typically OSPO, Security, or Product Compliance leads
2. Review quarterly to update status and note gaps
3. Integrate into your internal audit process or compliance dashboard
4. Use it as onboarding material for teams involved in CRA compliance

Domain & Objective	Key Checkpoints	Status	Driver & Notes
Governance & Ownership: Establish accountability and clear lines of responsibility for CRA compliance.	• CRA readiness owner assigned per product or business unit • Compliance responsibilities are defined across legal, engineering, security, and product teams • OSPO includes CRA compliance scope in the charter • A cross-functional steering group is formed for CRA oversight • Management regularly reviews CRA readiness progress	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
Policy & Awareness: Ensure all relevant teams understand CRA obligations and the impact on workflows	• Internal CRA policy is drafted and communicated • Training sessions held for developers, product managers, legal, and security teams • CRA obligations mapped to existing processes (security, compliance, DevSecOps) • CRA requirements are included in supplier and procurement agreements	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:

Domain & Objective	Key Checkpoints	Status	Driver & Notes
SBOM Management: Implement a reliable, automated SBOM process aligned with CRA requirements.	• Internal CRA policy is drafted and communicated • Automated SBOM generation is integrated into CI/CD pipelines • SBOM includes required component metadata (component name, version, supplier, license); vulnerability information is handled through defined vulnerability management processes • SBOM is validated for accuracy and completeness per release • SBOM storage and versioning is implemented • Access control and sharing procedures defined for regulators/ customers	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
OSS & Third-Party Dependency Oversight: Ensure OSS usage and third-party components are compliant and traceable.	• Open source policy updated for CRA compliance • SCA tools implemented and monitored • License and vulnerability data are regularly updated • Third-party suppliers required to provide SBOMs • Risk assessments are performed on relevant external dependencies based on product risk and criticality.	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
Vulnerability Management: Establish a structured process for identifying, reporting, and remediating vulnerabilities.	• Formal PSIRT process in place • Automated vulnerability scanning and alerting • Defined SLAs for remediation and disclosure • Secure vulnerability intake channel • CRA notification process defined for 24-hour reporting • Regular drills are conducted to validate response procedures	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:

Domain & Objective	Key Checkpoints	Status	Driver & Notes
Secure Development Lifecycle: Integrate security and compliance requirements throughout the development process.	• Secure coding standards are enforced • Dependency updates and patching are integrated into sprint cycles • Code signing and provenance tracking enabled • CRA requirement • Developers trained on SBOM and CRA best practices	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
Continuous Compliance & Monitoring: Move from periodic reviews to continuous evidence generation	• CI/CD pipeline automatically logs compliance artifacts (SBOMs, test reports, vulnerability scans) • Version control metadata linked to SBOMs • Compliance dashboard tracks CRA metrics across products • Deviations or non-compliance are automatically flagged • Regular internal audits are scheduled	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
Documentation & Audit Readiness: Ensure traceability and evidence for CRA audits.	• Central repository for CRA documentation established • Technical documentation includes SBOM, security tests, risk assessments, and vulnerability logs • Evidence retention policy defined in line with CRA technical documentation requirements (minimum 10 years where applicable) • Audit response procedures documented • Mock CRA audit performed to test readiness	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:

Domain & Objective	Key Checkpoints	Status	Driver & Notes
Cross-Functional Coordination: Align legal, technical, and business teams on compliance strategy.	• Regular CRA coordination meetings are held • Legal team validates interpretation of CRA obligations • Procurement includes CRA clauses in contracts • Customer communication strategy defined for security and SBOM disclosures • CRA-readiness metrics included in quarterly reporting	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes:
Continuous Improvement: Treat CRA as an evolving capability rather than a one-time project.	• Post-release vulnerability learnings reviewed • Annual CRA strategy review scheduled • Lessons from incident handling feedback into SDL • Emerging EU standards and guidance tracked • CRA maturity is assessed yearly	❑ Not Started ❑ In Progress ❑ Completed	Driver: Notes: