EDGEMTech

CYBER RESILIENCE ACT

Article 14 Readiness: A One-Hour Self-Check

Article 14 of the EU Cyber Resilience Act has applied since **11 September 2026.** It is the reporting duty: a manufacturer that becomes aware that a vulnerability in its product is being actively exploited has **24 hours** to file an early warning, **72 hours** for a notification with what is known by then, and **14 days** after a corrective measure exists for a final report. It applies to products **already on the market,** not only to new ones, and it outlives the support period.

Almost none of that is about vulnerabilities. Most of it is knowing what shipped, to whom, and who picks up the phone. This checklist takes about an hour, and you do not need anyone's help to run it.

How to score it honestly

Answer with evidence, not intention. "We have a process for that" scores zero. "Here is the file, and it is dated" scores one. There is no half mark: either you can show it today, or you cannot.

If finding the evidence takes more than ten minutes, it is a zero. The law counts in hours: 24 to warn, 72 to explain. You will not spend them searching.

	The question	What counts as a yes	Score
1	**Build identity.** Take a unit in the field. Can you derive the exact source it was built from?	The version string resolves to a specific commit of *every* repository involved, not only your own.	_____
2	**Archived software bill of materials (SBOM).** For each release in the field, can you open one now?	Produced by the build, not reconstructed from the image. Covers *every* image you ship.	_____
3	**Which unit runs what.** Which firmware version is on which customer's hardware today?	You can produce the list without asking the customer.	_____
4	**Rebuildability.** Pick a release from 18 months ago. Does it still build?	You have actually tried, recently. Never tested is a zero.	_____
5	**Component naming.** Does every SBOM component carry a name a vulnerability database recognizes?	Something checks this automatically and *fails* when a name matches nothing.	_____
6	**Warnings reach you.** Does a person or a system regularly check the vulnerability databases, the list of flaws under active attack, and warnings from your suppliers? And does your website say where to report a flaw?	You can open last week's list of new flaws found in your components, and say who read it. The address is a real security mailbox, not a contact form.	_____
7	**One way of deciding.** When a database flags one of your components, is there one written way to decide whether it really affects your product?	*Written down*, in a form you could hand a customer. It is also the legal test: not exploitable in your product, not reportable.	_____
8	**Named decision maker.** Who declares "we are aware" at 02:00 on a Sunday? Who is the deputy?	Both named in writing, both know it, the path tested once. "Aware" starts with the first assessment, not when someone finds time.	_____
9	**Fix and reach.** Can you build, test and ship a fix to deployed units, and tell affected users in time?	You have done it. Not that you could.	_____
10	**Filing route.** An EU Login account, registration started on the EU reporting platform (run by ENISA), and you know which national cyber security team would receive your report?	You can name that team and say why. For non-EU makers, Article 14(7) sets the order: representative, importer, distributor, users.	_____

Your score: _____ / 10

EDGEMTech

What the score means

- **0-3.** Normal, and not a scandal. We scored **2** the first time we ran this on one of our own projects.
- **4-7.** You have the engineering half. What is usually missing is 3, 8 and 9, which are not engineering problems and cannot be fixed by buying a tool.
- **8-10.** Unusual. Your remaining gap is almost certainly rehearsal: the difference between a path that exists and a path somebody has walked.

Five things that surprised us

- **An SBOM does not fail loudly. It answers confidently and wrongly.** A security-critical component was recorded under a name no vulnerability database has ever heard of. Wrong name → zero matches → looks clean, forever.
- **Zero results is the dangerous answer, not the safe one.** A wrong version returns plausible findings. A wrong name returns nothing at all, and nothing is what success looks like.
- **Scanning the finished product is no substitute for a list made during the build.** Scanning a shipped image found roughly one component in eight, and most of what it did find was operating system internals rather than the software that does the work. It then reported a confident, near-empty result.
- **Roughly half the vulnerability exposure we measured was a packaging question, not a patching backlog.** It came from software that did not need to be in the image at all. Half the surface leaves when the software leaves.
- **A fix can be invisible.** We closed an actively exploited vulnerability by upgrading the source, but the version string did not move, so every scanner on earth still reports it as open. That is what item 7 is for.

Two answers that do not count

"We have a documented process." Article 14 asks what you can produce, not what you have written down. The auditor question and the incident question are the same question.

"Our scanner says we are clean." Ours did too. Its vulnerability database had been dead since December 2023 (a retired feed returning an error), and it would have reported zero findings and a green build for over two years. Stock upstream code; nobody's fault, nobody's job. **Silence and safety are indistinguishable without someone whose job is telling them apart.**

Prepared by EDGEMTech SA, Yverdon-les-Bains, August 2026, from a readiness check we ran on our own work, and revised in September 2026 against the Commission's guidance on applying the regulation. It is a summary of the regulation as we read it, not legal advice. Yours to use, with no obligation attached.