Mend.io

SBOM COMPLIANCE GUIDE 2026

SBOMs that hold up.

The 2026 guide to compliance-ready software inventories

How to produce a software bill of materials that satisfies the CRA, CISA's 2026 Minimum Elements, PCI DSS, and your customers' security teams — and still answers "where are we exposed?" in minutes.



REQUIRE
What the rules now demand

BUILD
What a compliance-ready SBOM contains

OPERATE
The three jobs an SBOM must do

The inventory question got a deadline

For most of the last decade, "do you have an SBOM?" was a question security teams could answer with a spreadsheet and a promise. That stopped in 2026. The EU Cyber Resilience Act, CISA's 2026 Minimum Elements, and PCI DSS now say what an SBOM must contain, in which format, and how quickly it must feed an incident report.

The second shift is quieter but matters more. An SBOM that exists only to satisfy a request is a cost. An SBOM that your security team reaches for on the morning a zero-day drops is an asset. This guide is about building the second kind.

WHO THIS GUIDE IS FOR

AppSec and product security teams who own SBOM production, platform and DevOps engineers who wire it into the build, and the compliance and GRC leads who have to sign off.

WHAT THIS GUIDE COVERS

1. The 2026 requirement landscape: which rules apply to you and when
2. The anatomy of a compliance-ready SBOM: fields, formats, depth, VEX, cadence
3. The three jobs an SBOM must do: zero-day response, inventory and EOL, prioritization
4. Three artifacts to take with you: a 12-point readiness checklist, a customer-request playbook, and a zero-day response flow

CONTENTS

Three moves, one inventory

01 REQUIRE

Why SBOMs stopped being optional

1.1 Requirement map 03
Common threads 04

02 BUILD

What a compliance-ready SBOM contains

Generation points 05
2.1 Minimum data fields 06
VEX, supplier SBOMs 07

03 OPERATE

The three jobs an SBOM must do

Job 1 Zero-day response 08
Jobs 2 and 3 09

ARTIFACTS 4.1 TO 4.3

Checklist, request playbook, zero-day flow 10

SECTION 4

How Mend.io implements this 13

SECTION 1 · REQUIRE

The rules stopped asking whether you have an SBOM. They now say what goes in it.

01

ARTIFACT 1.1

The 2026 SBOM requirement map

REQUIREMENT	WHO IT APPLIES TO	WHAT IT ASKS OF YOUR SBOM	KEY DATES
EU Cyber Resilience Act (CRA)	Manufacturers placing products with digital elements on the EU market	Machine-readable SBOM covering at least top-level dependencies; 24-hour early warning on exploited vulnerabilities	Reporting from 11 Sep 2026; main obligations from 11 Dec 2027
CISA 2026 Minimum Elements	US federal software suppliers; the reference customers now cite	17 data fields across SBOM metadata and components; six practices	Published 29 Jul 2026; supersedes the 2021 NTIA list
PCI DSS 4.0.1, req. 6.3.2	Any organization in scope with bespoke or custom software	Maintain an inventory of bespoke and custom software and third-party components	Mandatory from 31 Mar 2025
FDA premarket guidance (524B)	Manufacturers of cyber devices submitting to the FDA	SBOM in premarket submissions, with component support status	Statutory since Mar 2023
Enterprise procurement	Any vendor selling to regulated or security-mature buyers	An SBOM per release, machine-readable, often with VEX	Per contract

24h
THE TEST

The 24-hour test

If a critical vulnerability in a widely used library were disclosed at 09:00 tomorrow, could you list every affected product, version, and customer by 09:00 the day after? If the honest answer is no, the rest of this guide is the path to yes.

Three things the 2026 rules have in common

01 **Machine-readable is the floor**

Every requirement in the map assumes a format a tool can parse. A PDF or spreadsheet inventory no longer qualifies. SPDX and CycloneDX are the two formats named consistently; the CISA 2026 elements dropped SWID tags from the list.

02 **Provenance matters as much as contents**

The 2026 Minimum Elements add fields the 2021 list never had: who generated the SBOM, with which tool and version, at what point in the lifecycle, and a signature to prove it. Regulators want to know not just what is in your software but how confident they should be in the answer.

03 **The SBOM feeds a clock**

The CRA's 24-hour early-warning window is impossible to meet from a stale inventory. When a vulnerability is disclosed, the SBOM is what tells you whether you have anything to report. An inventory produced once a year cannot start a 24-hour timer.

COMPANION GUIDES What this guide leaves to others

This guide stays on the software inventory. These topics have their own Mend.io guides.

License obligations, notice hygiene, model and open-weights licenses

The Complete Guide to Open Source & AI Licensing 2026

AI-BOM: inventorying models, agents, MCP servers, and AI frameworks

AI Security Governance: A Practical Framework

Securing AI agents, MCP servers & LLM apps

SECTION 2 · BUILD

Seventeen fields, two formats, one rule: generate it where the truth is.

02

Most SBOMs fail compliance review for boring reasons. They list components without versions, versions without identifiers, and identifiers without hashes. They cover direct dependencies and stop. They were generated from a manifest file rather than the artifact that shipped. This section sets the bar the 2026 requirements now expect.

Generate it where the truth is

Where you generate the SBOM decides what it can prove.

STEP 1 · SOURCE MANIFEST	STEP 2 · BUILD OR CI	STEP 3 · CONTAINER OR BINARY
SEES Declared dependencies	**SEES** Resolved dependency tree as built	**SEES** What actually ships, including OS packages
BLIND SPOTS Resolved versions, transitive tree, vendored code	**BLIND SPOTS** Runtime-injected components, base image contents	**BLIND SPOTS** Provenance of components without metadata
BEST FOR **Early developer feedback**	BEST FOR **The primary compliance SBOM**	BEST FOR **Delivered products, regulated submissions**

EARLIER IN THE LIFECYCLE → CLOSER TO WHAT SHIPS

THE PATTERN The compliance-ready pattern is a build-time SBOM enriched with a container or binary scan of the shipped artifact. Record the generation context field honestly for each.

The 2026 minimum data fields

Based on the CISA 2026 Minimum Elements. Fields marked NEW did not exist in the 2021 NTIA baseline.

FIELD	WHY IT MATTERS	2026
SBOM METADATA		
SBOM author	Who is accountable for the document	
SBOM author signature	Proves the SBOM wasn't altered in transit	NEW
Data format name and version	Lets the consumer's tooling parse it without guesswork	NEW
Generation context	Source, build, or binary analysis; sets confidence in completeness	NEW
SBOM timestamp	Ties the inventory to a point in time	
Tool name and version	Reproducibility; explains gaps when tooling changes	NEW
SBOM version	Distinguishes revisions of the same release's SBOM	NEW
COMPONENT		
Producer	Who published the component	
Name and version	The component and which build of it	
Identifiers (purl, CPE)	The key that joins to vulnerability data	
Hash value and algorithm	Confirms the exact artifact, not a lookalike	NEW
License	Declared license; obligations live in the Licensing 2026 guide	NEW
Dependency relationship	Which component pulled in which; makes transitive risk traceable	

THE SIX PRACTICES AUDITORS WILL ASK ABOUT

1. **Coverage** — transitive, not just declared
2. **Unknowns** — marked, never blank
3. **Frequency** — every release, every change
4. **Updates** — a revision path, not a rewrite
5. **Distribution** — predictable, access-controlled
6. **Machine-processable** — SPDX or CycloneDX

Attach exploitability, not just presence

An SBOM tells a customer what you ship. It does not tell them whether a listed vulnerability can hurt them. The Vulnerability Exploitability eXchange (VEX) closes that gap: a machine-readable statement, per vulnerability and product, of whether it is exploitable, not affected, or mitigated. CycloneDX carries VEX natively.

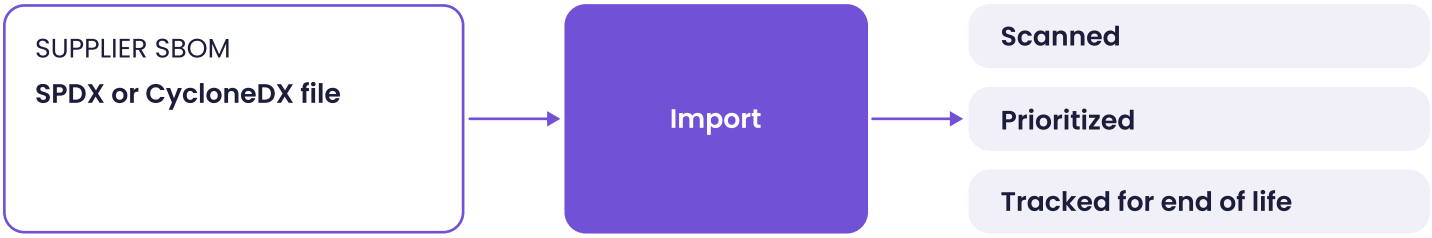
ONE SBOM, THREE VEX STATEMENTS illustrative

component-a 2.4.1	Vulnerable function called from a public endpoint	EXPLOITABLE
component-b 1.9.0	Vulnerable code path never reached	NOT AFFECTED
component-c 5.0.3	Compensating control in place	MITIGATED

THE PAYOFF Pair every SBOM you deliver with VEX data and you replace a week of customer questions with a field they can filter on.

Consume SBOMs, not just produce them

The CRA's due-diligence expectation runs upstream too. When you incorporate a third-party product, its SBOM is your fastest way to inventory what it brings. Choose tooling that can import a supplier's SPDX or CycloneDX file and treat those components like your own.



03

A compliant SBOM you never query is an expensive PDF.

Compliance is the reason most teams start an SBOM program. It is rarely the reason the program survives budget review. The programs that last are the ones where the SBOM does operational work every week. There are three jobs.

JOB 1
Zero-day response

JOB 2
Inventory and end of life

JOB 3
Prioritization

Job 1 · Zero-day response: answer "where are we exposed?" in minutes

When a headline vulnerability drops, the first question is not "how do we fix it?" but "do we have it, where, and is it reachable?" An SBOM indexed by component identifier turns that from a repository-by-repository hunt into a query.

THE PRACTICAL REQUIREMENTS

01 One index across every product
Every product and version in support. Not per team, not per repository. The question is organizational.

02 Transitive coverage
The vulnerable library is almost never a direct dependency. It is four levels down inside something else.

03 Reachability on top
Presence is not exposure. Knowing the vulnerable function is never called is what lets you deprioritize 30 findings and focus on the two that matter.

04 The output feeds the report
Under the CRA, the same query that tells you where you're exposed tells you whether you have a notifiable event and starts the 24-hour clock.

Job 2 · Inventory and end of life: know what you run before someone else tells you

An SBOM is also the only reliable record of what is aging out. Components reach end of life without a CVE. A maintainer walks away. A major version stops receiving patches. None of it appears in a vulnerability feed, and all of it is risk.

RUN IT AS A LIVING INVENTORY

- Flag components past end of life or with no release in a defined window; treat them like unpatched software
- Track which applications carry a component so one upgrade decision applies everywhere
- Record component support status in the SBOM you deliver; regulated buyers such as the FDA now expect it

Job 3 · Prioritization: make the backlog reflect risk, not counts

The SBOM gives you the denominator: every component, everywhere. Vulnerability data gives you the raw numerator. Neither tells you what to fix first. Prioritization is where an SBOM program earns its keep with developers, because it is what stops every finding from arriving as a ticket.

The signals worth engineering in, roughly in order of value

SIGNAL	QUESTION IT ANSWERS	SOURCE
1 **Reachability**	Is the vulnerable code path invoked by this application?	Static analysis of your code against the component
2 **Exploitability**	Is there a known exploit, and is the component exposed to untrusted input?	Threat intelligence plus deployment context
3 **Business context**	Is this application customer-facing, payments, or internal tooling?	Application classification
4 **Fix availability**	Is there a clean upgrade, and what does it break?	Dependency graph and release data

Close the loop: publish the resulting status back into your SBOM as VEX. That single step converts your internal prioritization into the exploitability statement your customers and auditors want to see.

SHIFT LEFT

SHIFT RIGHT

One SBOM serves both: generated in the pipeline so developers see component risk before merge, and queried in production so security can answer exposure questions about what is running today. If your SBOM only exists in one of those places, you have half a program.

SBOM readiness checklist

YOUR SCORE

__ / 12

Twelve checks to run against your current program. Score one point per yes.

☐	01	An SBOM is generated automatically for every release of every supported product	CRA Annex I · CISA frequency
☐	02	Generated at build time from the resolved dependency tree, with a container or binary scan for shipped artifacts	CISA generation context
☐	03	Includes transitive dependencies, not only direct ones	CISA coverage · CRA minimum exceeded
☐	04	Every component carries a standard identifier (purl or CPE) and a hash	CISA identifiers, hash
☐	05	The SBOM records its author, tool, tool version, and generation context	CISA metadata fields
☐	06	Unknown or undeterminable fields are marked as unknown, not left blank	CISA unknown information
☐	07	Exported in SPDX or CycloneDX in a version your largest customers accept	CISA machine-processable · CRA
☐	08	Vulnerability exploitability status is attached as VEX	Customer and auditor expectation
☐	09	Component support and end-of-life status is tracked and, where required, included	FDA 524B · inventory practice
☐	10	One organization-wide index lists every product affected by a component in under an hour	CRA 24-hour early warning
☐	11	Supplier SBOMs are imported and their components tracked like your own	CRA due diligence
☐	12	A named owner can deliver the SBOM for any supported version on request, with access control	CISA distribution · CRA

INTERPRETING YOUR SCORE

0 – 4

An inventory problem before a compliance problem. Start with checks 1 to 3.

5 – 8

You can produce an SBOM but not defend it. Focus on checks 4 to 8.

9 – 12

Your SBOM is doing operational work. The remaining checks are about delivery and evidence.

The customer SBOM-request playbook

A security questionnaire or procurement email asks for your SBOM. Five steps from request to delivery.

48h
TARGET

1. **Confirm scope**
   Which product, which version, and which format and version do their tools accept? Ask once; do not guess. Most requests resolve to CycloneDX JSON or SPDX JSON for a named release.

2. **Pull, don't build**
   Export the SBOM for that release from your platform. If you have to assemble it by hand, note that as a gap for check 1 of the readiness checklist.

3. **Attach exploitability** THE STEP THAT SAVES THE MOST
   Include VEX status for every listed vulnerability so their team can filter to what is exploitable in your product. This is the step that prevents the follow-up email listing 40 CVEs.

4. **Sign and version it**
   Apply the author signature and an SBOM version. Record who received which version and when.

5. **Set the cadence**
   Tell them how they will receive updates: on each release, on request, or via a portal. Put it in the contract language if it isn't already.

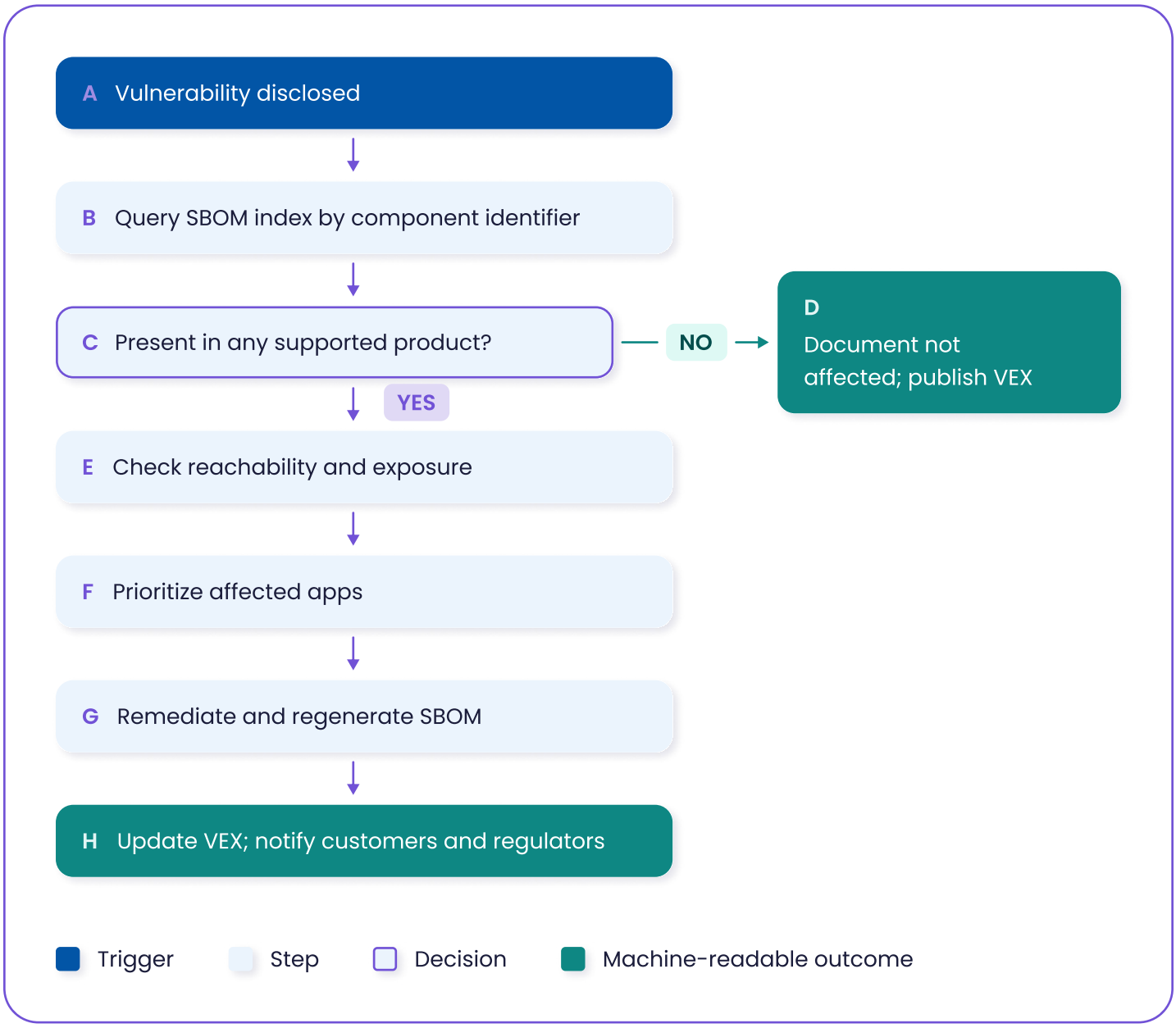
WHAT NOT TO SEND

- An SBOM for a different version than the one they run
- An SBOM without a timestamp
- A spreadsheet

Each one generates more work than it saves.

Zero-day response with an SBOM

A reference flow. The SBOM query is the first step, not the last.



Read top to bottom: the SBOM query is the first step, not the last, and every branch ends in a machine-readable statement your customers can consume.

Everything in Sections 1 to 3 can be built by hand. The question is whether it stays current when your teams ship daily.

Mend.io generates, enriches, and operates SBOMs as part of Mend AppSec, so the compliance artifact and the operational inventory are the same data.

75%
less remediation effort for teams using Mend.io

80%
faster remediation at WTW with Mend.io

FRAMEWORK NEED	HOW MEND.IO COVERS IT
Machine-readable export Section 2	Mend SCA exports the Dependencies SBOM report in SPDX 2.2 and 2.3 and CycloneDX 1.4, 1.5, and 1.6 at project, application, or organization scope
Generate where the truth is Section 2	Build-time scanning of the resolved dependency tree via the Mend CLI and CI integrations; container image scanning produces SBOMs for what ships, including OS packages
Exploitability, not presence Sections 2 and 3	CycloneDX exports carry VEX status per vulnerability, distinguishing exploitable, unreachable, and mitigated findings, driven by reachability analysis
Consume supplier SBOMs Section 2	Import a CycloneDX or SPDX file to create or update a project, so third-party components are scanned and tracked alongside your own
Zero-day response Job 1	One inventory across every project answers "where is this component?" organization-wide; reachability separates presence from exposure
Prioritization Job 3	Reachability-driven SCA and contextual application classification rank findings by real risk, with unified remediation recommendations
Remediation	Mend Renovate opens dependency-update pull requests with confidence signals so fixes land in the developer's flow
Governance and evidence	Policy enforcement in the pipeline and audit-ready reporting; the same SBOM and VEX data serve auditors, regulators, and customers

CLOSING

Modern application risk doesn't live in one component. It lives in the tree of dependencies you didn't choose and the versions you didn't notice changing.

An SBOM that meets the 2026 requirements and answers exposure questions the same day is how you prove control of both. Mend.io gives you one inventory that does both jobs.

Get a demo See how Mend.io generates compliance-ready SBOMs.

SOURCES

Cyber Resilience Act: reporting obligations
European Commission

CRA explained: scope, classes, deadlines
cyberresilienceact.eu

2026 Minimum Elements for a Software Bill of Materials
CISA

CISA minimum elements field breakdown
sbomify

PCI DSS v4.0.1 software catalog mandate
HALOCK

Cybersecurity in Medical Devices premarket guidance
FDA

The Dependencies SBOM Report
Mend.io documentation

SBOM Report for Container Images
Mend.io documentation

Mend SCA release notes
Mend.io documentation

Mend Platform release notes
Mend.io documentation

Mend.io is built for every risk, across AI and AppSec. By securing the code layer and the AI layer — and the interactions between them, where modern application risk now lives — **Mend.io** extends proven AppSec workflows to the models, prompts, and agents inside today's applications, delivering continuous protection across the entire AI application lifecycle.

© 2026 Mend.io | All rights reserved

Visit our website