

SBOM Vulnerability Management for Medical Devices (524B)

Reviewed 2026-10-22 | Turn a medical device SBOM into a working vulnerability management program: PURL hygiene, CVE matching, safety-weighted triage, VEX, and postmarket monitoring.

Managing SBOM vulnerabilities under Section 524B means providing the SBOM as required submission content for covered cyber devices and using it as an input to an ongoing process: give every component a resolvable identifier, match continuously against CVE feeds, triage by patient safety impact rather than CVSS alone, and document device-specific exploitability assessments. VEX is an optional way to communicate those assessments, not an FDA-required format. The FDA reviews this monitoring process premarket, and gaps may prompt questions.

How to turn an SBOM into a working vulnerability management program: identifier hygiene, CVE matching, safety-weighted triage, VEX, and the postmarket loop the FDA expects to see before clearance.

SBOM VULNERABILITY MANAGEMENT · UPDATED 2026

Most manufacturers can now produce an SBOM. Far fewer can show what happens on the Tuesday a critical CVE lands in a library that ships in their device. That gap is what the FDA is checking, and it is what a security researcher will find first.

What Section 524B requires for SBOM vulnerability management

[Section 524B](#) requires a cyber device manufacturer to provide an SBOM to the FDA (the Secretary) and a postmarket vulnerability plan. Section 524B(b)(2) calls for a reasonably justified regular patch cycle and patches for critical vulnerabilities presenting uncontrolled risk as soon as possible out of cycle, without numeric SLAs. The [February 3, 2026 premarket guidance](#) adds the expectations that make this operational:

- The guidance recommends machine-readable SBOMs and component support level and end-of-support dates, which may be in a separate addendum (§V.A.4(b), p. 17).
- Known vulnerabilities in listed components should be assessed for exploitability and patient safety impact, not merely enumerated.
- The plan for continued monitoring after clearance is reviewed **premarket**, and inadequate plans may prompt review questions.

Step 1: Fix identifiers before anything else

Automated matching benefits from [PURL or CPE](#). Where unavailable, use names, versions, supplier advisories, CVE-ID queries, and documented manual review.

Where identifiers usually go missing:

- **Vendored and forked source** copied into the firmware tree with no package manager record.
- **Statically linked C libraries**, the classic embedded blind spot.

- **Board support packages and RTOS components** delivered by silicon vendors.
- **Renamed or repackaged upstream projects** whose CPE differs from the name you use internally.

Run a coverage metric: percentage of components with a resolvable identifier. A 95 percent target is illustrative, not an FDA threshold; assess unresolved gaps by risk.

Step 2: Match continuously, not at submission

Feed the SBOM against a CVE source on a fixed cadence, for example weekly, per build, or on feed update, with the cadence justified by risk rather than an FDA numeric minimum. Useful sources:

- [NVD](#) for the CVE baseline
- [CISA Known Exploited Vulnerabilities catalog](#) for what is being used in the wild
- [CISA ICS medical advisories](#) for device-specific disclosures
- Upstream project advisories, which frequently precede NVD by days

Match on version ranges, not names. Record every match with the date it appeared so you can prove response time later.

Step 3: Triage on patient harm

This is the step where IT-derived programs fail in MedTech. CVSS measures impact to an information system. It does not measure harm to a patient.

A workable triage model:

Tier	Criteria	Target response
P1	Exploitable and can affect therapy delivery, monitoring accuracy, or device availability	Immediate assessment, fix, or compensating control on an expedited path
P2	Exploitable and exposes PHI or enables unauthorized access without direct safety impact	Fix in the next scheduled release
P3	Exploitable only with physical access or an implausible precondition	Scheduled, documented
P4	Not exploitable in this configuration	Document evidence and manufacturer disposition; optionally communicate through VEX

Two things make this defensible: written criteria agreed before the first incident, and a clinical or systems engineer, not only a security engineer, judging the safety column.

Step 4: Use VEX to close what is not exploitable

A typical device firmware image produces hundreds of CVE matches. Some may not be reachable: the vulnerable function is never called, the affected feature is compiled out, or the attack requires a configuration you do not ship.

VEX lets you state that formally, with a status and a justification, so customers and reviewers get an answer instead of a raw list. The four statuses are not affected, affected, fixed, and under investigation. The justification is what makes it credible, so name the reason: component not present, vulnerable code not in the execute path, inline mitigation already applied.

CycloneDX carries VEX natively. SPDX 2.3 pairs with external VEX such as CSAF; SPDX 3.0 supports native VEX in its Security profile. VEX is not an FDA-required format.

Step 5: Fix, and prove the fix reached the field

Patching is where medical devices differ hardest from IT. An update must be validated, may require regression testing against clinical function, and sometimes needs a regulatory assessment before release.

Your process needs:

- A **change assessment** that determines whether the fix requires a new submission or falls under an existing [PCCP](#)
- **Validation** proportionate to the risk of the change
- A **deployment path** with signature verification and rollback
- **Field verification** that the update actually installed across the installed base
- **Records** connecting CVE, decision, validation, and deployment

Devices that cannot be patched need documented compensating controls and customer communication. That is a legitimate outcome; silence is not.

Step 6: Close the loop to the QMS

Vulnerability management is not a parallel universe to quality. Connect it:

- Externally reported vulnerabilities enter through your [CVD process](#) and are recorded as complaints where applicable.
- Systemic causes, such as an unmaintained upstream component, go to CAPA.
- Vulnerabilities with a safety consequence update the ISO 14971 risk file.
- Anything meeting the reporting threshold follows your recall and reporting procedures.

Reviewers and auditors both look for these links. A vulnerability program that never touches the QMS reads as informal.

Two recurring SBOM monitoring failure modes

Transitive dependency gaps. Teams inventory direct dependencies and stop. The vulnerable component is usually three levels down, inside a library your library depends on. Generate the SBOM from the build so the full tree is captured, and check the depth of your dependency relationships.

Static SBOMs. An SBOM generated once for the submission is stale within weeks. We recommend updating it for component changes and field releases, with each version retained so you can answer which components shipped in which firmware version two years from now.

What to include in the premarket submission

- The monitoring cadence and the CVE sources you use
- Your triage criteria and SLAs, written down
- The exploitability assessment method and evidence; your VEX approach, if used
- The patch validation and deployment path, with expected timelines for critical findings
- The linkage to complaint handling, CAPA, and recall procedures
- Evidence the process is running today, not planned

Worked example: the Tuesday a critical CVE lands

Abstract process descriptions are easy to write and easy for a reviewer to discount. This is the sequence we expect a functioning program to be able to evidence, using a heap overflow in a compression library shipped inside an infusion pump's telemetry service.

Day 0, 14:10. The feed job runs and matches pkg:generic/zlib@1.2.13 against a newly published CVE with a CVSS base of 8.1. A ticket opens automatically with the CVE, the matched component, the affected firmware versions from the SBOM archive, and the timestamp of first detection. Record awareness as well as detection. The December 28, 2016 postmarket guidance's 30/60-day provisions are a conditional Part 806 enforcement-discretion policy measured from awareness, not a universal patch clock. Reporting duties arise under 21 CFR Parts 803 and 806 when applicable, not Section 524B.

Day 0, 16:00. Reachability triage. A security engineer checks whether the vulnerable code path is compiled in and reachable. Here it is: the telemetry service inflates a payload received over the hospital network before parsing it. A documented not-affected assessment could support manufacturer closure, with VEX optional; exploitable moves it to safety review.

Day 1. Safety review with a systems engineer who owns the pump's essential performance. The finding cannot alter a dose, but a crash loop in the telemetry service stalls the alarm relay to the nurse station. That is an availability failure inside a monitoring path, so it lands at P1 despite being neither remote code execution nor a PHI exposure.

Day 2. Change assessment. The fix is a library bump with no change to the therapy algorithm or the labeling, so the team evaluates whether it falls within any applicable authorized PCCP or other change-control pathway. If it had required altering the alarm logic, this is the point where a new submission assessment starts, and that determination gets written down either way.

Days 3 to 12. Patch, regression test against the clinical function, sign the image, and validate the update path on representative hardware, including rollback.

Days 13 to 45. Staged deployment, then field verification: what percentage of the installed base is actually running the fixed version. This number is where most programs stop measuring, and it is the one that determines whether the vulnerability is really closed.

Throughout. A single record ties CVE to detection timestamp, reachability rationale, safety determination, change assessment, validation evidence, deployment date, and field coverage. That record supports review but does not guarantee an audit outcome.

Metrics that prove the vulnerability program is running

A reviewer or auditor is testing whether the process is operational, not whether it is documented. The following metrics and numeric targets are illustrative internal examples, not FDA thresholds or acceptable residual-risk levels. Adapt them to patient risk and do not use closure rates to force dispositions.

Metric	How to compute	Where it should sit
Identifier coverage	Components with a resolvable PURL or CPE divided by total components	Above 95 percent
Feed latency	Median hours from CVE publication to ticket creation	Under 48 hours
Triage latency	Median days from ticket creation to safety determination	Under 5 days for P1
Assessment coverage	Matched vulnerabilities with documented device-specific assessments	Track completeness; no target for not-affected dispositions
Remediation age	Oldest open P1 and P2 findings, by device family	No P1 older than the SLA you published
Field coverage	Installed units on the fixed version divided by affected units	Trending to 90 percent within two release cycles

Track them per device family, not per company. An aggregate that averages a well-maintained cloud service against a neglected legacy pump hides the exact problem the metric exists to surface.

Where embedded builds leak SBOM components

Coverage gaps in device firmware are structural, and they repeat across manufacturers:

- **Statically linked C libraries.** No package manager touched them, so nothing records them. Binary composition analysis on the shipped image is the only reliable detection.
- **Vendor board support packages.** Silicon vendors ship a tree that contains an RTOS, a TCP/IP stack, and crypto code, often several versions behind upstream. If the supplier's SBOM is thin, inventory it

yourself and record who authored what.

- **Bootloaders and recovery images.** Frequently excluded from the SBOM because they are built by a separate job, and frequently the most security-relevant code on the device.
- **Firmware for peripheral microcontrollers.** A pump with a separate motor controller has a second SBOM that nobody generated.
- **Build-time toolchain artifacts** that end up embedded in the binary, such as a libc variant baked in at link time.
- **Cloud companion containers** where scanning the application layer skips the base image entirely.

A useful audit question: take the release image, run binary composition analysis against it, and diff the result against your published SBOM. The delta is your real gap, and it is usually between 5 and 30 components on a first pass.

How Blue Goat Cyber runs this

We stand up the pipeline end to end: identifier remediation so matching actually works, continuous monitoring against NVD, KEV, and vendor advisories, safety-weighted triage reviewed by engineers who understand the clinical function, optional VEX statements supported by device-specific assessments, and QMS-linked records so it holds up in an audit. Delivered as [postmarket SBOM and VEX monitoring](#), or as part of a [premarket package](#).

Where this fits in the cluster

- [CycloneDX vs. SPDX for Medical Devices](#)
- [CPE vs. PURL in Medical Device SBOMs](#)
- [VEX Documents for Medical Devices](#)
- [Postmarket Cybersecurity Monitoring Program](#)
- [Coordinated Vulnerability Disclosure for Medical Devices](#)
- [FDA SBOM requirements](#) - the fields the FDA expects in every component record.

Related from Blue Goat Cyber

- [Postmarket SBOM & VEX Monitoring](#)
- [FDA Postmarket Cybersecurity Services](#)
- [FDA Premarket Cybersecurity Services](#)

FAQ

How do I use an SBOM for medical device vulnerability management?

Treat the SBOM as the asset inventory that drives a continuous process: ensure every component has a PURL or CPE, match it against CVE feeds on a fixed cadence, triage matches by exploitability and patient safety impact, document non-exploitability with device-specific evidence, optionally using VEX, patch what is, and record all of it in your QMS.

What are the FDA requirements for SBOM in postmarket surveillance?

Section 524B requires monitoring, identifying, and addressing postmarket vulnerabilities on a reasonably justified schedule, and providing the SBOM to the FDA (the Secretary). User availability is a labeling recommendation, not the Section 524B(b)(3) obligation. The February 3, 2026 guidance expects support level and end-of-support data per component and a documented monitoring, triage, and patch process, which is reviewed before clearance.

How does VEX work with medical device SBOMs?

VEX pairs with the SBOM to state whether each known CVE actually affects your product, using one of four statuses plus a justification. It is an optional communication format for device-specific assessments, not a substitute for supporting evidence or manufacturer risk decisions. The FDA does not require VEX or prescribe a not-affected percentage.

How often should medical device SBOMs be updated for vulnerabilities?

We recommend updating the SBOM for component changes and field releases. Weekly or daily scans are illustrative internal targets, not FDA minimums; justify the cadence and provide urgent escalation.

What tools are used for SBOM vulnerability scanning in MedTech?

Common generators include Syft, CycloneDX build plugins, and Yocto or Buildroot SBOM output for embedded Linux. Common matchers include Gripe, OSV-Scanner, and Dependency-Track, which also stores VEX. Tooling matters less than identifier coverage and a triage process that accounts for patient harm.

How do I handle vulnerabilities in third-party libraries I cannot patch?

Document it. Assess exploitability in your configuration, apply compensating controls such as disabling the affected feature or restricting the interface, record the residual risk with a benefit-risk rationale, communicate to customers if there is a safety-relevant consequence, and track the upstream fix. Documented rationale supports assessment but does not establish acceptability or guarantee a review outcome.

Sources & primary references

- [Cybersecurity in Medical Devices: Quality Management System Considerations and Content of Premarket Submissions \(FDA, February 3, 2026 final guidance\)](#)
- [Section 524B of the FD&C Act](#)

- [The Minimum Elements For a Software Bill of Materials \(NTIA\)](#)
- [CISA Known Exploited Vulnerabilities Catalog](#)
- [National Vulnerability Database \(NIST\)](#)
- [FDA Postmarket Management of Cybersecurity in Medical Devices](#)

Free tools for this

- [SBOM Readiness Checker](#) - check an SBOM against the FDA's expected fields before you submit it.

Current web guide: <https://bluegoatcyber.com/guides/sbom-vulnerability-management-medical-devices>

Informational material. Confirm requirements for your device and market. The manufacturer owns risk decisions, QMS approvals, and regulatory submissions. No regulatory outcome is promised.