BLUE GOAT CYBER

Medical Device Cybersecurity · FDA Specialists

FDA SBOM + VEX PLAYBOOK

The FDA-Compliant SBOM + VEX Playbook

How to generate, validate, and continuously update a CycloneDX SBOM with VEX statements that survives FDA review and powers postmarket.

bluegoatcyber.com

© Blue Goat Cyber. For educational use. Not legal or regulatory advice.

WHY THIS MATTERS

The FDA-compliant SBOM + VEX playbook.

Under the FDA's 2026 final premarket cybersecurity guidance, every cyber device submission must include a **machine-readable SBOM** in SPDX 2.3+ or CycloneDX 1.4+ format, plus an ongoing process for monitoring vulnerabilities in those components. PDF SBOMs and spreadsheets are no longer acceptable as the SBOM itself. This playbook gives you a working blueprint for generating, validating, and continuously updating an SBOM that survives FDA review and scales into your postmarket program.

Key takeaway: An SBOM without VEX is noise. VEX (Vulnerability Exploitability eXchange) is what turns 'this CVE matched a component' into 'and here's whether it's actually exploitable in our device' - which is the question reviewers and customers actually ask.

1. SPDX vs. CycloneDX: which one?

SPDX 2.3+	Linux Foundation standard. Strong on license compliance metadata. Widely accepted by FDA reviewers.
CycloneDX 1.5	OWASP standard. Native first-class support for VEX, services, ML-BOM, SaaS-BOM, formulation. **Our default for medical devices.**
Recommendation	Use CycloneDX 1.5 with embedded VEX. If your downstream tooling requires SPDX, generate both - the source of truth should be CycloneDX.

2. What MUST be in your SBOM

- Every third-party dependency (direct + transitive)
- Operating system / RTOS and kernel version
- Bootloader, firmware blobs, and signed binary components
- Open-source libraries statically linked into firmware
- Container base images and layers (for SaMD with cloud back-ends)
- Hardware components with embedded firmware (where the manufacturer's SBOM is available)
- Build-time toolchain components if they ship in the artifact (tracers, embedded compilers)
- PURL (Package URL) identifiers for every component - reviewers use these to map CVEs
- Cryptographic hashes (SHA-256 minimum) for binary components
- License identifiers in SPDX form

3. Generation strategy by device type

SaMD / cloud SaaS	Build-time SCA (Syft, cdxgen, npm/pip/maven plugins). Generate per-build, store with release artifacts.
Mobile companion app	Combine build-time SCA for iOS/Android with platform-specific scanners (CocoaPods, Gradle dependency reports).
Embedded / firmware	Build-time SCA + binary analysis (Binwalk, Ghidra-assisted extraction, BLINT). Reconcile with bill-of-materials from the build system.
Hybrid device + cloud	Generate one SBOM per deployable artifact (device firmware, mobile app, cloud service). Link them in the submission narrative.

4. VEX: turning CVE noise into reviewable evidence

Once your SBOM is in monitoring, you'll get a daily firehose of CVE matches. Most are not actually exploitable in *your* device - the vulnerable code path may be unreachable, compensating controls may be in place, or the affected feature may be disabled. VEX is the standardized way to communicate this, both internally and to FDA reviewers and customers.

VEX status values (CycloneDX)

not_affected	The vulnerable code is present but cannot be exploited in this device's configuration. Must include a justification (e.g., *vulnerable_code_not_in_execute_path*).
affected	The device is exploitable. Triage and patch.
fixed	The device was affected; a patch has shipped in version X.
under_investigation	Triage in progress. Time-bound - escalate if it sits here for >30 days.

Acceptable not_affected justifications

- *component_not_present* - the vulnerable subcomponent isn't shipped
- *vulnerable_code_not_present* - we ship the library but not the vulnerable file
- *vulnerable_code_not_in_execute_path* - present but never reached
- *vulnerable_code_cannot_be_controlled_by_adversary* - reachable but not influenceable
- *inline_mitigations_already_exist* - sandbox/seccomp/network ACLs neutralize the threat

5. The 30-day SBOM operating cadence

- **Every build:** regenerate SBOM, diff against previous, fail the build if any new **critical** component appears without review.
- **Daily:** automated CVE scan against the latest SBOM. Auto-create tickets for High+.
- **Weekly:** security engineer triage. Set VEX status. Close out resolved.
- **Monthly:** publish updated SBOM + VEX to customers per your CVD policy.
- **Quarterly:** SBOM hygiene review - drop end-of-life components, update PURLs, validate licenses.

6. What FDA reviewers actually look for

- Machine-readable file in the submission package (not embedded as PDF text)
- Component-level CVE assessment with CVSS scores and VEX status
- Evidence of a **process**, not a one-time snapshot - sample VEX statements, monitoring tool screenshots, ticket trails

- Tie-in to your postmarket cybersecurity management plan
- Disclosure plan for downstream operators (hospitals, IDNs) - typically aligned to MDS^2 form

Key takeaway: If a reviewer can't reproduce your CVE list by feeding your SBOM into an open scanner, your SBOM isn't compliant - regardless of how thorough the PDF appendix looks.

READY FOR THE NEXT STEP?

Talk to a senior medical device security engineer.

We've contributed cybersecurity documentation, threat models, SBOMs, and pen test reports to **250+ FDA submissions** across 510(k), De Novo, and PMA pathways - with a **100% submission success rate**. If you're building a cyber device and want to go in with airtight artifacts on the first try, we should talk.

Book an SBOM working session
30 minutes. No slide deck. Bring your device profile and your top three questions.

bluegoatcyber.com/contact

WHAT WE DO

- **FDA premarket cybersecurity** - SPDF, threat model, SBOM, pen test, security risk assessment, eSTAR-ready package
- **FDA cybersecurity deficiency response** - rapid remediation when reviewers send a letter
- **Postmarket programs** - vulnerability monitoring, VEX, coordinated disclosure, patch validation
- **Legacy device remediation** - bring older devices up to current FDA expectations