



BLUE GOAT CYBER
Medical Device Cybersecurity

MEDICAL DEVICE CYBERSECURITY GUIDE

FDA Cybersecurity Testing Requirements: 2026 Taxonomy

Every cybersecurity test the FDA's Feb 2026 guidance expects premarket, as a 10-family taxonomy mapped to our eSTAR attachment groups and recognized standards.

Reviewed 2026-10-22

bluegoatcyber.com

Book a free discovery call
30 minutes with a senior expert
bluegoatcyber.com/discovery-call



IN BRIEF

The FDA's February 3, 2026 guidance, pp. 25-26, recommends security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing. The ten families below are BGC's organizing framework for techniques, not ten FDA-defined categories. Independent internal testers may be used; third-party testing is not universally required.

BGC's ten-family organizing framework maps testing techniques to the FDA's four recommended categories and to our evidence packaging approach.

The FDA recommends four testing categories. This guide organizes supporting techniques into BGC's ten-family framework; applicability depends on the device, its threats, and documented testing rationale.

Last updated: June 2026 · Aligned to the FDA's [February 2026 final guidance](#) and [Section 524B of the FD&C Act](#).

TL;DR: BGC's 10-family organizing framework

1. **Static analysis (SAST):** source-level vulnerability and defect detection.
2. **Software composition analysis (SCA) / SBOM vulnerability analysis:** known-CVE triage of third-party components, with optional VEX.
3. **Dynamic application security testing (DAST):** runtime scanning of network-reachable interfaces.
4. **Vulnerability scanning:** network, host, and configuration baseline scanning.
5. **Fuzz testing:** protocol-aware fuzzing of medical and transport protocols.
6. **Penetration testing:** manual, threat-model-driven, exercising hardware, radio, firmware, and business logic.
7. **Abuse and misuse case testing:** negative requirements, malformed inputs, intended-misuse scenarios.
8. **Patch and update mechanism testing:** signed updates, rollback, root-of-trust integrity.
9. **Authentication, authorization, and access control testing:** RBAC, service accounts, session, key handling.
10. **Cryptographic implementation testing:** algorithm choices, key lifecycle, FIPS 140-3 validation status.

Every one of these must trace back to the [threat model](#) in attachment group 3 and forward into the residual-risk argument in Group 4. **Testing without traceability is testing the reviewer cannot credit.**

How the FDA frames "security testing" in the 2026 guidance

The FDA's February 3, 2026 final guidance recommends security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing (pp. 25-26). Section 524B establishes statutory obligations; the guidance provides nonbinding recommendations for supporting evidence.

[BGC SUMMARY, NOT A QUOTATION] Document the four recommended categories, including applicable vulnerability testing techniques, and relate the evidence to requirements, threats, and risk decisions.

We package testing evidence as **Group 7 (Testing)** within the eSTAR v7.1 cybersecurity section, with supporting evidence in Group 5 (SBOM) and Group 6 (Controls). These groups are our packaging, not eSTAR field names. For the full map, see our [eSTAR v7.1 mapping guide](#).

The rest of this page walks each of the 10 families.

1. Static analysis (SAST)

What it is. Automated analysis of source code (or compiled bytecode/IR) to surface known defect patterns: buffer overflows, taint flows, injection sinks, insecure API usage, hardcoded secrets, weak random number generation, dangerous compiler flags.

What it catches. Defects that are visible in code structure: input → sink flows, memory-safety patterns in C/C++, deserialization sinks, hardcoded credentials, weak crypto primitives, dangerous functions (strcpy, system, eval).

What it misses. Anything that depends on runtime state or external configuration: authentication logic bugs, business-logic abuse, race conditions in distributed systems, vulnerabilities in third-party binaries you don't have source for, and every hardware or radio attack path.

Example tools, not FDA-endorsed selections. Coverity, Fortify, Checkmarx, Semgrep, SonarQube, CodeQL, MISRA-aware tools (LDRA, Polyspace) for safety-critical embedded code.

Where it lives. Attachment group 7 (Testing). The SAST report should cover the first-party code base with documented scope, rule set, suppression rationale for every "not applicable," and a remediation log.

Common deficiency pattern. "Provide SAST results for the [module] code base and rationale for suppressed findings." This usually means the submission included SAST output for one module but not the connected-device firmware, or included a tool summary with no suppression log.

Standards anchor. IEC 81001-5-1 §5.6; NIST SSDF PW.7; AAMI SW96:2023 §6.

2. Software composition analysis (SCA) / SBOM vulnerability analysis

What it is. Identification of third-party and open-source components in the device (the SBOM), correlated against known-vulnerability feeds (NVD, GitHub Advisory Database, vendor advisories, CISA KEV, EPSS), with a device-specific vulnerability assessment. An optional **VEX** (Vulnerability Exploitability eXchange) statement may communicate status: not_affected, under_investigation, affected, or fixed.

What it catches. Known CVEs in third-party components, the single largest source of post-clearance vulnerabilities. EOL/end-of-support components. License-derived risk where it overlaps with security (unmaintained projects).

What it misses. Zero-days, custom first-party code (that's SAST/DAST), and configuration mistakes when a component is integrated insecurely.

Example tools, not FDA-endorsed selections. Anchore, Snyk, Dependency-Track, Black Duck, Mend (WhiteSource), Trivy, Grype, Syft. The FDA recommends a machine-readable SBOM and industry-accepted formats, but specifies neither CycloneDX nor SPDX as mandatory and no format version floor.

Where it lives. SBOM itself in attachment group 5; the vulnerability analysis and VEX statements live in Group 7 (Testing). See our [SBOM vulnerability management guide](#) and [CycloneDX vs SPDX comparison](#).

Common deficiency pattern. An SBOM without device-specific vulnerability applicability and risk assessments may leave evidence gaps. VEX is optional, not required by Section 524B(b)(3).

Standards anchor. Section 524B(b)(3); NTIA minimum elements; AAMI TIR97:2019/(R)2023 for postmarket triage.

3. Dynamic application security testing (DAST)

What it is. Automated runtime scanning of network-reachable interfaces: HTTP/HTTPS endpoints, REST/GraphQL/SOAP APIs, and some network services, performed without source-code access.

What it catches. OWASP Top 10 and API Top 10 patterns (injection, XSS, SSRF, broken auth, misconfiguration, outdated components), TLS configuration issues, weak ciphers, expired certs, default credentials reachable over the network.

What it misses. BLE, USB, JTAG, UART, SPI, firmware, cellular, NFC, MedRadio, business-logic abuse, and any multi-step exploit chain. DAST scopes itself to whatever a scanner can reach over a network.

Example tools, not FDA-endorsed selections. Burp Suite Professional, OWASP ZAP, Acunetix, Invicti (Netsparker), Nuclei.

Where it lives. Attachment group 7. The report must document scope (which endpoints were tested), authentication state (unauthenticated + each role), the scan profile or rule set, findings, and the remediation log.

Common deficiency pattern. Submissions that present a DAST report **as** the penetration test. Reviewers issue a deficiency requesting evidence that testing covered the device's actual interfaces. See our deeper post: [DAST vs penetration testing](#).

Standards anchor. OWASP ASVS / MASVS; IEC 81001-5-1 §9.7.

4. Vulnerability scanning

What it is. Network-, host-, and configuration-level scanning to baseline the device and its environment against known CVEs and misconfigurations. Distinct from DAST in that it scans the **system**, not the **application logic**.

What it catches. Missing OS patches on the device and any companion systems, dangerous open ports and services, weak service banners, default OS credentials, insecure network protocols (Telnet, SMBv1, anonymous LDAP), TLS/SSH version and cipher posture.

What it misses. Application-layer bugs (DAST/pen test territory), hardware, radio, and any vulnerability not yet in a public feed.

Example tools, not FDA-endorsed selections. Nessus, OpenVAS / Greenbone, Qualys VMDR, Rapid7 InsightVM, Nuclei (with vulnerability templates).

Where it lives. Attachment group 7, often bundled into the penetration test report appendix.

Common deficiency pattern. Vulnerability scan output included without remediation log or suppression rationale; high-severity findings present with no disposition.

Standards anchor. NIST SP 800-115; ISO/IEC 27001 Annex A.12.

5. Fuzz testing

What it is. Generation of malformed, unexpected, or randomized inputs to a protocol or interface to find crash, hang, memory-corruption, and state-confusion bugs. For medical devices, this almost always means **protocol-aware fuzzing** (generic byte-flipping does not exercise DICOM, HL7, FHIR, or BLE GATT meaningfully).

What it catches. Memory-safety bugs in protocol parsers, state-machine confusion in pairing or session setup, denial-of-service paths, malformed-input crashes that translate directly into patient-safety hazards.

What it misses. Anything that requires understanding intent: authorization bugs, business-logic abuse, missing controls. Fuzzing finds crashes, not policy gaps.

Example tools, not FDA-endorsed selections. AFL++, libFuzzer, Honggfuzz, Boofuzz, peach3, Defensics (Synopsys), Mayhem (ForAllSecure), medical-protocol-specific fuzzers for DICOM, HL7 v2, FHIR, ASTM, and BLE GATT. See our [fuzz harness generation guide](#).

Where it lives. Attachment group 7, inside the penetration test report or as a separate fuzz testing appendix.

Common deficiency pattern. "Provide evidence of fuzz testing against [DICOM/HL7/FHIR/BLE] interfaces." Generic HTTP fuzzing does not satisfy this when the device speaks medical protocols.

Standards anchor. IEC 81001-5-1 §9.7; AAMI SW96:2023 §6.4.

6. Penetration testing

What it is. Methodology-driven, manual security testing by a qualified independent assessor, scoped to the device's threat model, exercising every interface the device actually exposes: network, radio (BLE, Wi-Fi, cellular, NFC, MedRadio), USB, debug ports (JTAG/SWD/UART/SPI), firmware update mechanism, and business-logic abuse. Penetration testing **uses DAST, SAST, SCA, vulnerability scanning, and fuzzing as inputs** and extends well beyond them.

What it catches. Multi-step exploit chains, hardware tamper paths, firmware downgrade and signature-bypass paths, radio-protocol attacks, USB service-mode abuse, JTAG/UART firmware extraction, business-logic abuse, side-channel and fault injection where the threat model requires.

What it misses. Things by design out of scope. If your engagement statement of work excludes a class of testing, that exclusion needs an explicit residual-risk argument.

Example tools, not FDA-endorsed selections. Tooling is not the bar; the **named, independent assessor with documented qualifications** is. Tools we see in accepted reports: Burp Suite Pro, Metasploit, Nessus, Bettercap, Ubertooth/HackRF/Proxmark/Flipper Zero, Saleae logic analyzers, ChipWhisperer, JTAGulator, and protocol-specific fuzzers.

Where it lives. Attachment group 7 as the signed penetration test report, with a traceability matrix mapping every Group 3 threat to the test(s) that exercised it.

Common deficiency pattern. A DAST report presented as the pen test (see family 3). Web-only pen test on a Bluetooth device. Testing without documented independence from device development. Independent internal testers may be appropriate; a third party is not universally mandatory.

Standards anchor. Section 524B; AAMI SW96:2023 §6.5; IEC 81001-5-1 §9.7; OWASP MASVS for SaMD; NIST SP 800-115.

For depth: [FDA penetration testing requirements](#) · [scoping a medical device penetration test](#) · [BLE/RF pen test case study](#).

7. Abuse and misuse case testing

What it is. Testing against **negative requirements** (what the device must not do), derived from the threat model and from intended-misuse scenarios (IEC 62366-1 human-factors abuse cases), including malformed inputs, unexpected user actions, off-spec environments, and adversarial workflow chaining.

What it catches. Behavior the requirements never explicitly forbade but that creates safety or security hazards: workflow chaining (combining two legitimate actions into an unsafe sequence), input-validation gaps that aren't strict-protocol-fuzz territory, intended-misuse paths a clinician or patient could take, off-spec environments (low battery, network drop, sensor failure mid-procedure).

What it misses. Implementation defects you'd find with SAST or pen testing. Abuse cases are about **scenario coverage**, not bug discovery.

Example tools, not FDA-endorsed selections. No single tool; this is methodology-driven. Common approaches: threat-model-derived abuse-case worksheets, exploratory testing logs, IEC 62366-1 use-error tables extended with adversarial paths.

Where it lives. Attachment group 7, often cross-referenced from Group 2 (Risk Management) and Group 3 (Threat Model) because abuse-case results feed both human-factors and security risk files.

Common deficiency pattern. Reviewers see a long list of positive test cases and no evidence of negative-requirement testing. "Provide test evidence for abuse/misuse cases identified in the threat model."

Standards anchor. AAMI TIR57:2016/(R)2023 / ANSI/AAMI SW96:2023 5.6; IEC 62366-1; IEC 62304 5.5. See [abuse and misuse case testing](#).

8. Patch and update mechanism testing

What it is. Verification that the device can be updated "in a reasonably and reliably timely manner," the statutory phrasing from **Section 524B(b)(2)**. Tests the signed-update path end-to-end: signature verification, root-of-trust integrity, rollback handling, anti-rollback (version monotonicity), update-channel authentication, failure-mode safety, and time-from-CVE-to-deployed-patch SLA evidence.

What it catches. Unsigned or weakly-signed updates, missing rollback handling, downgrade attacks that re-enable patched vulnerabilities, broken root-of-trust chains, update mechanisms that brick the device on a failed transfer.

What it misses. The vulnerabilities themselves; this family proves you can **deliver fixes**, not that the fixes exist.

Example tools, not FDA-endorsed selections. Custom test harnesses against the update endpoint, signing-key compromise simulation, version-rollback test cases, network-interruption tests during update.

Where it lives. Attachment group 6 (Controls) describes the mechanism; Group 7 (Testing) holds the evidence.

Common deficiency pattern. "Provide evidence of patchability demonstrating compliance with Section 524B(b)(2)." Submissions describe a mechanism but show no test results exercising it.

Standards anchor. Section 524B(b)(2); NIST SP 800-193 (Platform Firmware Resiliency); IEC 81001-5-1 §9.4.

9. Authentication, authorization, and access control testing

What it is. Verification of identity, role enforcement, and key-handling for every account class on the device: clinician, patient, service technician, manufacturer, machine-to-machine. Covers RBAC enforcement, session management, multi-factor where required, account lockout, password policy, service-account isolation, and key lifecycle.

What it catches. Authentication bypass, privilege escalation, horizontal authorization gaps (IDOR), broken session handling, hardcoded service credentials, weak default credentials, missing rate limits on credential endpoints.

What it misses. Implementation defects in the underlying crypto (family 10), and any attack that doesn't go through the auth surface.

Example tools, not FDA-endorsed selections. Burp Suite Pro intruder/repeater, Postman/curl-based test suites, OWASP ASVS-aligned manual test plans, automation via custom scripts.

Where it lives. Group 6 (Controls) describes the design; Group 7 (Testing) holds the evidence.

Common deficiency pattern. "Provide test evidence for role-separation enforcement between [role A] and [role B]." Or: hardcoded service credential discovered in pen test → traced to missing auth test plan coverage.

Standards anchor. IEC 81001-5-1 §6.1, 9.7; OWASP ASVS 2 (authentication), 4 (access control); NIST SP 800-63B.

10. Cryptographic implementation testing

What it is. Verification that cryptography is correctly chosen, correctly implemented, and correctly operated across its lifecycle. Covers algorithm selection, key generation entropy, key storage (HSM/TPM/secure element/keystore), key rotation, certificate handling, TLS configuration, and **FIPS 140-3 validation status** where applicable.

What it catches. Weak or deprecated algorithms (MD5, SHA-1 in signatures, DES, RC4), insecure modes (ECB), weak key generation (low entropy, predictable seeds), missing certificate validation, downgrade-vulnerable TLS configuration, key material exposed in logs or memory, expired or self-signed certs in production paths.

What it misses. Application-layer bugs that bypass crypto entirely (e.g., an API that accepts both a JWT and a static API key).

Example tools, not FDA-endorsed selections. testssl.sh, sslyze, nmap NSE crypto scripts, custom test harnesses for key-rotation flows, FIPS 140-3 module certificate references from NIST CMVP.

Where it lives. Group 6 (Controls) describes the cryptographic posture; Group 7 (Testing) holds the evidence.

Common deficiency pattern. "Provide FIPS 140-3 validation status for the cryptographic module used for [function]" or "Provide test evidence that key rotation does not interrupt clinical function."

Standards anchor. FIPS 140-3; NIST SP 800-131A (algorithm transitions); NIST SP 800-57 (key management).

Traceability is its own deliverable

Every test result in families 1-10 must trace back to the threat model in attachment group 3 and forward into the residual-risk argument in Group 4. Reviewers organize their assessment by **threat**, not by tool. A scanner report organized by URL is not a traceable artifact until you map each finding to the threats it exercises.

A clean traceability matrix has, at minimum: threat ID · test family · test ID(s) · finding(s) · remediation or residual-risk argument · evidence pointer (file/page). Most accepted submissions we have seen format this as a single spreadsheet appendix to the Testing attachment.

How the 10 families fit together

A premarket submission for a connected medical device that **passes Group 7 review on first read** typically includes:

- **SAST report** (family 1): first-party code base, with suppression log.
- **SBOM and vulnerability assessments** (family 2): industry-accepted machine-readable formats are recommended; VEX is optional.
- **DAST report** (family 3): every network-reachable interface, authenticated and unauthenticated.
- **Vulnerability scan output** (family 4): typically appended to the pen test report.
- **Fuzz testing evidence** (family 5): medical-protocol-aware where the device speaks medical protocols.
- **Independent penetration test report** (family 6): signed by a named, independent assessor, scoped to the threat model.
- **Abuse/misuse case test evidence** (family 7): negative-requirement testing.
- **Patch/update mechanism test results** (family 8): evidence supporting Section 524B(b)(2).
- **Auth/access control test plan + results** (family 9): every role, every account class.
- **Cryptographic implementation test results** (family 10): including FIPS 140-3 status where applicable.
- **Traceability matrix**: threat → test → result → residual-risk argument.

This is company planning guidance, not an automatic deficiency rule. Select techniques and evidence based on device risks and clearly justify coverage.

How Blue Goat Cyber covers the 10 families

We scope every medical-device cybersecurity engagement against this 10-family taxonomy. Static and composition analysis run in the build pipeline; dynamic and vulnerability scanning run against the deployed environment; fuzzing runs against the actual protocols the device speaks (DICOM, HL7, FHIR, BLE GATT, USB, ASTM); penetration testing happens on a calibrated bench with the real hardware in front of a qualified, independent assessor (CISSP / OSCP / ex-military red team); abuse-case, patch-mechanism, auth, and crypto testing layer in based on the threat model. We deliver the full Group 7 package (every report, the traceability matrix, and the residual-risk argument) formatted for direct eSTAR upload.

If the FDA raises a cybersecurity question, we answer it at no extra cost until the cybersecurity portion is closed. See our [penetration testing service](#) and the companion [eSTAR v7.1 mapping guide](#).

FAQ

Does the FDA require every one of these 10 testing families on every submission?

Not literally every family on every device; a pure cloud-hosted SaMD with no hardware doesn't need JTAG/UART pen testing inside family 6, and a device with no BLE doesn't need BLE fuzzing inside family 5. But every family that **could apply given the device's threat surface must either be performed or have an explicit residual-risk argument for why it was excluded**. Silent exclusion is a deficiency pattern.

What's the difference between vulnerability scanning and DAST?

Vulnerability scanning baselines the **system**: OS patch level, open ports, service banners, network-protocol posture. DAST tests the **application**: endpoint behavior, input handling, authentication logic, business logic. Both live in Group 7 and both are usually expected on any device with network-reachable services.

Is fuzz testing required if we already ran pen testing?

The FDA lists fuzz testing within vulnerability testing, one of four recommended categories. Consider protocol-aware fuzzing appropriate to interfaces and risks. Evidence may be consolidated in a report or appendix; a separately filed fuzz-testing report is not prescribed.

Can we use a single tool that "does it all"?

No tool covers all 10 families, and reviewers know this. Be skeptical of any vendor pitch claiming a single platform satisfies the entire Testing attachment. The portfolio matters more than the brand on any individual report.

What's the most common Group 7 deficiency?

Across the deficiency letters we have seen since the February 2026 final guidance, the top three patterns are: (1) a DAST report presented as the penetration test (family 3 vs 6 confusion), (2) SBOM with missing device-specific vulnerability assessments (family 2 gap; VEX is optional), and (3) missing fuzz evidence for the medical protocols the device speaks (family 5 gap on a device with BLE, DICOM, HL7, or FHIR).

Where does threat modeling fit into the testing taxonomy?

Threat modeling is **not** a testing family; it lives in Group 3 as its own deliverable. But every entry in the 10 families must trace back to it. Without a Group 3 threat model, the test results in Group 7 have nothing to map to and reviewers cannot credit coverage. See our [STRIDE threat modeling guide](#).

Related reading

- [eSTAR v7.1 Cybersecurity Attachments mapping guide](#)
- [FDA cybersecurity guidance](#)
- [FDA Premarket Cybersecurity Submission Checklist](#)
- [DAST vs Penetration Testing](#)
- [FDA Penetration Testing Requirements](#)
- [FDA Security Requirements Testing: The 2026 Verification Guide](#)
- [Scoping a Medical Device Penetration Test](#)
- [Fuzz Harness Generation for Medical Device Protocols](#)
- [Abuse and Misuse Case Testing](#)
- [SBOM Vulnerability Management](#)

- [STRIDE Threat Modeling for Medical Devices](#)

References

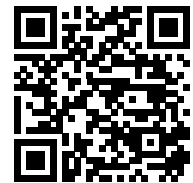
Primary sources cited or relied on in this guide:

- [Cybersecurity in Medical Devices: Quality Management System Considerations and Content of Premarket Submissions \(Feb 2026 final\)](#), U.S. FDA
- [ANSI/AAMI SW96:2023, Standard for Medical Device Security](#), AAMI/ANSI
- [IEC 81001-5-1:2021, Health software and health IT systems safety, effectiveness, and security](#), IEC/ISO
- [OWASP Application Security Verification Standard \(ASVS\)](#), OWASP
- [Section 524B of the FD&C Act \(Consolidated Appropriations Act, 2023, §3305\)](#), U.S. Congress

Current web guide: bluegoatcyber.com/guides/fda-cybersecurity-testing-requirements-taxonomy

Talk to a senior medical device cybersecurity expert

Book a free 30-minute discovery call. Scan the code or visit bluegoatcyber.com/discovery-call. You get a written quote within 24 hours.



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