



BLUE GOAT CYBER
Medical Device Cybersecurity

MEDICAL DEVICE CYBERSECURITY GUIDE

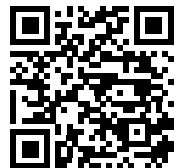
FDA Security Requirements Testing: Verification Guide (2026)

What security requirements testing means in the FDA's Feb 2026 guidance, how boundary analysis works, and why it is verification you own.

Reviewed 2026-10-22

bluegoatcyber.com

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



IN BRIEF

Security requirements testing is documented evidence that each security design input requirement was implemented successfully, including boundary analysis and the rationale behind the boundary assumptions used. It is verification against your own design inputs under your own quality system, so the manufacturer owns it. Penetration testing is adversarial security testing of the built device by independent testers who may review those design inputs. The FDA's February 3, 2026 final premarket guidance recommends evidence covering both categories, with clear coverage of their distinct purposes. One combined report may address multiple categories when the evidence supports them.

A penetration test alone may leave gaps in security requirements verification, threat mitigation evidence, and vulnerability testing. Qualified suppliers may execute documented verification; the manufacturer retains responsibility.

Aligned to the FDA's [February 2026 final premarket cybersecurity guidance](#) and [Section 524B of the FD&C Act](#).

Why this matters

Section 524B(b)(2) of the FD&C Act requires manufacturers to design, develop, and maintain processes that provide a reasonable assurance the device and related systems are cybersecure. The February 3, 2026 final guidance, *Cybersecurity in Medical Devices: Quality Management System Considerations and Content of Premarket Submissions*, is the operative interpretation reviewers apply, and it frames security testing as evidence generated inside a Secure Product Development Framework rather than as a one-time external assessment.

That framing has a consequence teams underestimate. Security testing evidence is quality system output. It is traceable to design inputs, produced under the QMSR, effective February 2, 2026, incorporating ISO 13485 design and development requirements, and reviewed by the FDA in the same frame as any other verification and validation record. When a submission contains only an outside report, the reviewer sees an assessment with no anchor in the design history file, and the natural deficiency follows: show us the requirements, show us they were implemented, show us how you tested the boundaries.

The four testing categories the guidance expects

Section V.C recommends documentation addressing these four testing categories, not four separately filed deliverables:

Category	Question it answers	Who can produce it
Security requirements testing	Was each security design input requirement implemented successfully, and what boundaries were tested?	Manufacturer or qualified supplier under controlled procedures
Threat mitigation testing	Does each risk control from the threat model actually work on the built device?	Manufacturer, with third-party support on specific controls
Vulnerability testing	What known and discoverable weaknesses exist in the build?	Manufacturer tooling plus third-party testing
Penetration testing	What can a capable adversary achieve against the released configuration?	Independent testers, organizationally separate from the design team

GUIDANCE SUMMARY

Address all four categories clearly. They may be documented in one report or several, provided coverage and traceability are explicit.

What security requirements testing actually is

Security requirements testing is verification in the classic design-controls sense. You wrote security requirements as design inputs. You built the device. Now you demonstrate, requirement by requirement, that the built device implements each one as specified.

Concretely, the evidence package contains:

- **A list of the security design input requirements**, each with a stable identifier that appears in your traceability matrix.
- **A test case or set of test cases per requirement**, with the pass criteria written in advance.
- **Results tied back to the requirement identifiers**, so a reviewer can walk from any requirement to its evidence without asking you for a mapping.
- **Boundary analysis**, plus the rationale for the boundary assumptions used.
- **Disposition of anything that did not pass**, either fixed and retested, or carried into the residual-risk argument with justification.

The requirements themselves come from your threat model, your security risk assessment, and the control expectations in the guidance: authentication, authorization, cryptography, secure update and rollback, logging and detection, resiliency and recovery, and the interface-specific controls your architecture views describe.

Boundary analysis, and why the rationale matters more than the numbers

Boundary analysis is the part of this deliverable that separates a credible package from a checkbox exercise, and it is where most teams under-document.

Every security requirement has implicit limits. A session timeout requirement has a minimum and maximum. A key length requirement has an accepted floor. A rate-limit control has a threshold that separates normal clinical use from abuse. A buffer has a maximum accepted input size. Requirements testing has to exercise the values at and around those limits: at the boundary, one step inside, one step outside, and at the invalid extremes.

The guidance goes further than asking for the test results. It asks for **the rationale behind the boundary assumptions**. That is a claim about your device's intended use, not about your code. It says: this is the maximum realistic number of concurrent clinical users, so this is the rate limit; this is the longest a clinician can be away from the console mid-procedure, so this is the session timeout; this is the largest legitimate payload this interface receives from the hospital system, so this is the input size ceiling.

A reviewer evaluating that rationale is checking whether your boundaries were derived from the clinical workflow and the threat model, or picked because they were the framework defaults. The second one draws questions.

GUIDANCE SUMMARY

Testing documentation should include evidence that each security requirement derived from the design inputs was implemented successfully, including boundary analysis and the rationale for the boundary assumptions used.

Manufacturer responsibility and supplier execution

Outsourcing execution does not transfer manufacturer responsibility.

Security requirements testing verifies the device against documented design inputs under controlled procedures. A qualified supplier can receive the requirements, protocols, traceability information, and boundary rationale needed to execute and document verification.

The honest division of labor:

- **The manufacturer** owns the security requirements, the traceability matrix, the boundary assumptions and their clinical rationale, the verification protocols, and the signed verification records.
- **An external testing partner** can build test harnesses, execute protocol-level and interface-level tests against nominated boundaries, and report findings as inputs to your verification records. What the partner signs is its own scope, method, and results.
- **Qualified suppliers** may execute documented verification within an agreed scope; manufacturer review and acceptance remain subject to its quality system.

This is not a limitation to work around. It is the reason the deliverable exists: the FDA wants to see that security was engineered into design controls, not inspected in afterwards.

Penetration testing: the other end of the spectrum

Penetration testing asks a different question. Not "did we build what we specified," but "what can someone do to what we built."

It is adversarial, it is threat-model-driven rather than requirements-driven, and it deliberately looks for behavior nobody specified. The 2026 guidance expects the report to document tester independence and expertise, scope, duration, methods, and results. Independence matters because a tester who helped write the requirements will unconsciously test inside them, which is precisely the blind spot the exercise is meant to cover.

The most valuable findings from a good penetration test are the ones that were never in the requirements at all: a chained path that combines two individually acceptable behaviors, a debug interface nobody documented, a maintenance credential shipped in the field image. No amount of requirements verification surfaces those, because requirements verification only proves the specified thing works.

For scope and depth, see the [penetration testing guide](#) and the [510\(k\) pen test scoping guide](#).

Threat mitigation testing sits between them

Threat mitigation testing is the bridge, and it is the category teams skip most often.

Your threat model identifies threats and assigns risk controls. Threat mitigation testing demonstrates that each of those controls is effective on the built device. It is not "the requirement was implemented" (that is requirements testing) and it is not "an adversary tried everything" (that is penetration testing). It is: this specific control was intended to mitigate this specific threat, and here is the evidence that it does.

The traceability chain reviewers follow runs threat, to risk control, to design input requirement, to verification evidence, to mitigation effectiveness evidence. An unexplained gap in that chain may prompt a request for additional information. See the [threat modeling guide](#) for the upstream half.

How to file all four in eSTAR

Section V.C recommends evidence covering all four categories. You may use one combined report or several reports with clear cross-references. Distinct sections, methods, results, and traceability help reviewers locate coverage; the guidance does not require separate files or state that a combined report causes rejection.

Practical structure:

1. A one-page testing summary that names all four categories and points to where each is filed.
2. Security requirements testing: the traceability matrix, the verification protocols and results, the boundary analysis with rationale.
3. Threat mitigation testing: control-by-control effectiveness evidence, cross-referenced to the threat model.

4. Vulnerability testing: SAST, DAST, software composition analysis, fuzzing, and scanning results, with the SBOM and VEX cross-references.
5. Penetration testing: the independent report, with the independence statement, scope, duration, methods, and results.
6. A remediation and residual-risk table covering findings from every category.

For the full slot mapping, see the [eSTAR v7.1 cybersecurity mapping guide](#) and the [10-family testing taxonomy](#).

How Blue Goat approaches this

We scope third-party work to what a third party can legitimately attest to. Penetration testing, vulnerability testing, protocol fuzzing, and control-effectiveness testing are engagements we run and sign, with our own scope, methodology, tester independence statement, and results.

As company policy, Blue Goat Cyber does not perform design verification or validation. We provide penetration and security testing, while the manufacturer arranges requirements verification internally or through qualified suppliers. This narrower service scope is not an FDA prohibition on outsourcing.

FAQ

Can our penetration test report satisfy the security requirements testing expectation?

Not by itself. Requirements verification needs documented coverage of design inputs and boundary analysis. A combined report can support multiple categories if it clearly documents the distinct methods, results, and traceability.

Who signs the security requirements testing records?

Records are executed, reviewed, and approved as specified by the manufacturer's quality system and supplier agreements. Qualified suppliers may sign execution records; the manufacturer retains responsibility for acceptance.

What does the FDA mean by boundary analysis?

Testing at, inside, and outside the limits of each security requirement: minimum and maximum accepted values, thresholds, timeouts, key lengths, input sizes, and rate limits. The guidance also asks for the rationale behind the boundary assumptions, meaning the justification, grounded in intended use and the threat model, for why those limits are the right ones for this device.

Do we need requirements testing if we have full SAST and DAST coverage?

Yes. SAST and DAST are vulnerability testing. They find defect patterns and runtime weaknesses, but neither one knows what your security requirements say, so neither can demonstrate that a specific requirement was implemented as specified. They are complementary evidence in a different category.

How much boundary rationale is enough?

Enough that a reviewer can follow the reasoning without asking a question. One or two sentences per boundary that connect the value to the clinical workflow, the intended use environment, or a specific threat in the model. Boundaries stated with no rationale, or justified only as framework defaults, are the version that draws Additional Information requests.

What if a security requirement fails verification?

Fix it and retest, or document it as an accepted residual risk with a justification tied to the security risk assessment. What you cannot do is omit it. Reviewers who find a requirement in your traceability matrix with no corresponding result treat the gap as unresolved, and an unexplained gap is more damaging than a documented and justified one.

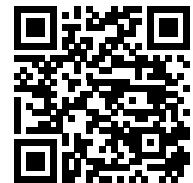
CTA

If your submission timeline is close and you are unsure whether your testing evidence covers all four categories the guidance expects, [book a review](#). We will tell you which deliverables are ours to run and which are yours to sign.

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

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.