



GUIDE

# 12 Critical Findings from Medical Device Penetration Tests

*Free Pen-Test Guide · Updated 2026 · FDA-Aligned 12 Critical Findings from Medical Device Penetration Tests A practical, ungated guide to the most common high- and critical-severity findings we surface*

Free Pen-Test Guide · Updated 2026 · FDA-Aligned

A practical, ungated guide to the most common high- and critical-severity findings we surface in medical device pen tests, what each one looks like in the field, and exactly how to fix it before FDA or an attacker finds it first.

## Overview

- 250+ - Submissions Cleared
- 12 - Most Common Findings
- 6-8w - Full-Scope Engagement
- 0 - FDA Rejections (Context)

## Why pen testing is now a gating activity, not a checkbox

FDA's 2026 Premarket Cybersecurity Guidance and Section 524B make penetration testing an explicit submission element for cyber devices. Reviewers expect testing that exercises every interface, with a methodology that matches the device's threat model, performed by testers who understand medical-device context. A narrow web-only report is now its own deficiency.

The patterns repeat. Across hundreds of medical device assessments, the same twelve findings show up again and again regardless of vendor, modality, or class. This guide walks through each one: what it looks like in the field, why it matters to patients and reviewers, and exactly how to fix it.

## How to use this guide

- Read each finding as a self-assessment: have you tested for it on production hardware, not just dev kits?
- Use the checklist at the end to score your pen-test readiness before you scope an engagement.
- Where you find a gap, the fix list is concrete enough to start work on Monday.



## | What we find - and how to fix it

Every finding below is drawn from real medical device assessments. Each one is a potential FDA deficiency and a patient-safety risk.

### **Exposed debug and service interfaces (UART, JTAG, SWD)**

CRITICAL Hardware / Firmware

What we see in the field

Production devices ship with active UART consoles, unfused JTAG/SWD headers, or vendor service ports that grant root or memory access with no authentication.

Why it matters

Physical-access attacks are explicitly in scope under FDA's threat model expectations. An open debug port lets an attacker pull firmware, extract keys, or pivot to the patient network in minutes.

How to fix it

- Disable or fuse JTAG/SWD before manufacturing release; document the fuse state in production records.
- Authenticate or remove UART consoles; never leave a root shell on a production unit.
- Treat every physical interface as an attack surface in the threat model and pen test scope.

### **Firmware extractable and not encrypted at rest**

CRITICAL Hardware / Firmware

What we see in the field

Firmware can be pulled from flash via SPI clip, debug interface, or update file, and decompiles cleanly to readable code, strings, and embedded credentials.

Why it matters

Once firmware is extractable in plaintext, every other control degrades: secrets leak, update logic can be reversed, and clones become trivial. FDA reviewers expect a documented firmware protection strategy.

How to fix it

- Enable secure boot and flash encryption on the MCU/SoC; verify it on production hardware, not just dev kits.
- Remove embedded credentials, API keys, and private keys from firmware images.
- Validate that update files are encrypted in transit and at rest where feasible.



## **Update mechanism without signature verification or rollback protection**

CRITICAL Firmware / Update Path

What we see in the field

Firmware updates are accepted without cryptographic signature checks, signatures use weak keys, or downgrade attacks succeed against unpatched versions.

Why it matters

The update path is the single most dangerous interface on a connected device. An unsigned or downgrade-vulnerable updater turns one compromised endpoint into a fleet-wide incident.

How to fix it

- Enforce signature verification with a hardware root of trust; reject any unsigned image.
- Implement anti-rollback counters so older, vulnerable firmware cannot be reinstalled.
- Test the update path end-to-end including failure, interruption, and tampering scenarios.

## **Default, hardcoded, or shared credentials**

CRITICAL Authentication

What we see in the field

Service accounts, BLE pairing PINs, web admin logins, or cloud API tokens are identical across every unit, or default passwords are never forced to change.

Why it matters

Hardcoded credentials are the #1 finding in connected medical device assessments and a frequent FDA deficiency. One leaked credential compromises the entire installed base.

How to fix it

- Generate per-device credentials at provisioning; store the root secret in secure element or TPM.
- Force credential rotation on first use for any user-facing account.
- Eliminate hardcoded API keys and certificates from firmware and mobile companion apps; use static analysis (SAST) to detect them.

Worried about hardware or firmware findings?

A 30-minute call with a senior tester can scope the right assessment.

Talk to an expert



## **Wireless stack misconfiguration (BLE, Wi-Fi, cellular)**

HIGH Wireless

What we see in the field

BLE pairing uses Just Works, characteristics are world-readable/writable, Wi-Fi falls back to WPA2-PSK with shared keys, or cellular APNs lack mutual authentication.

Why it matters

FDA expects testing across every wireless interface with an appropriate methodology. Weak pairing or open characteristics let an attacker on the same RF range read patient data or control therapy.

How to fix it

- Use BLE LE Secure Connections with numeric comparison or OOB pairing; restrict GATT permissions.
- Require WPA2-Enterprise or WPA3 for Wi-Fi where the use environment supports it.
- Mutually authenticate cellular and cloud connections with device certificates, not shared secrets.

## **TLS implemented but not validated**

HIGH Network / Cloud

What we see in the field

Device uses TLS but accepts any certificate, pins to an expired CA, supports deprecated ciphers (RC4, 3DES, TLS 1.0/1.1), or skips hostname verification.

Why it matters

A TLS connection that doesn't validate the server is functionally cleartext to anyone on the path. Reviewers and pen testers both check this on every cloud interface.

How to fix it

- Pin to a defined CA or specific certificate; reject everything else.
- Enforce TLS 1.2+ with modern cipher suites; disable legacy protocols at the library level.
- Validate hostname, expiry, and revocation; fail closed when validation fails.

## **Cloud API authorization gaps (IDOR, missing tenant checks)**

CRITICAL Cloud / API

What we see in the field

Authenticated users can access other patients' data, devices, or organizations by changing an ID in the URL or request body.

Why it matters



Insecure direct object reference (IDOR) and broken tenant isolation are the most common findings in medical device cloud back-ends, and the most damaging when exploited.

How to fix it

- Enforce authorization on every endpoint; never trust client-supplied IDs without an ownership check.
- Add tenant scoping at the data layer (RLS, query filters) so a missing check fails closed.
- Cover IDOR explicitly in API pen testing with multiple test accounts across tenants.

## **Companion mobile app leaks secrets or weakens device security**

HIGH Mobile

What we see in the field

iOS/Android app contains hardcoded API keys, ships in debug mode, lacks certificate pinning, or stores PHI in unencrypted local storage.

Why it matters

The mobile app is part of the device's attack surface. A reverse-engineered app often hands an attacker the keys to the cloud back-end and every paired device. See the OWASP Mobile Top 10.

How to fix it

- Strip secrets from the app; use short-lived tokens fetched after authentication.
- Implement certificate pinning and runtime integrity checks; ship release builds only.
- Encrypt local PHI using platform keystores; clear sensitive data on logout and uninstall. Use SAST to catch secrets early.

Want a defensible pen test report FDA already recognizes?

We do fixed-fee engagements with medical-device specialists.

Our pen test services

## **Insufficient logging and tamper-evident audit trail**

HIGH Monitoring / Post-Market

What we see in the field

Device and cloud lack logs for security-relevant events (auth, config changes, firmware updates), or logs can be modified or deleted without detection.

Why it matters

Without trustworthy logs you cannot investigate an incident, satisfy post-market monitoring expectations, or prove that a control worked. FDA's post-market guidance assumes you can.

How to fix it



- Log authentication, authorization, configuration, and update events with timestamps and identifiers.
- Forward logs to an append-only sink; alert on gaps and integrity failures.
- Define retention aligned to incident response and regulatory needs.

## **Denial-of-service paths that affect patient safety**

CRITICAL Availability / Safety

What we see in the field

Malformed BLE/USB/network input crashes the device, locks the UI, or forces a reboot mid-therapy; no watchdog or safe-state behavior.

Why it matters

A DoS on a medical device isn't an availability nuisance - it's a potential harm event. Reviewers expect both fuzz testing and a documented safe-state behavior under attack.

How to fix it

- Fuzz every external interface (BLE, USB, Wi-Fi, cloud APIs) and fix crashes, not just symptoms.
- Implement a watchdog and a defined safe state for unexpected faults.
- Tie DoS scenarios into the ISO 14971 risk file with documented mitigations.

## **Vulnerable third-party components with no monitoring plan**

HIGH SBOM / Post-Market

What we see in the field

SBOM lists components with known CVEs, or there's no process to learn about new vulnerabilities in the libraries shipped on the device.

Why it matters

Most exploitable findings in medical device pen tests come from outdated open-source components. FDA expects you to monitor them throughout the device lifecycle - not just at submission. Check NVD and CISA advisories.

How to fix it

- Scan the SBOM against NVD and vendor advisories on every build and on a recurring schedule.
- Define risk-based SLAs for triage and patching; document them in the post-market plan.
- Track end-of-life components and plan migrations before support runs out.



## **Pen test scope that misses real attack surface**

CRITICAL Scope / Methodology

What we see in the field

Test report covers only the web interface or only the device, leaves out cloud APIs, mobile app, BLE, USB, or service tools, or relies on automated scanning alone.

Why it matters

FDA's premarket guidance is explicit that testing must exercise the full system. A narrow report is a deficiency on its own, regardless of what the testing found.

How to fix it

- Scope across device hardware, firmware, wireless, cloud APIs, mobile companion app, and service tools.
- Combine manual exploitation with automated tooling; pure scans are not pen testing.
- Engage testers with documented medical-device experience and provide them the threat model.

## **Pen-Test Readiness Checklist**

If you can't check all twelve, you have known gaps a tester - or an attacker - will surface. Use this as a go/no-go before you book your assessment.

## **Pen-Test Timeline Reality Check**

Teams routinely underestimate how long a credible medical device pen test takes. Here's what a full-scope engagement typically looks like for a moderate-complexity Class II connected device:

WorkstreamTypical Effort

Hardware and firmware analysis2 to 3 weeks

Wireless testing (BLE, Wi-Fi, cellular)1 to 2 weeks

Cloud API and back-end assessment1 to 2 weeks

Mobile companion app testing1 week

Note: Workstreams run partially in parallel. Most engagements complete in 6 to 8 weeks; rework after a critical finding typically adds 2 to 4 weeks of remediation and retest.

## **When to bring in a specialist pen-test partner**

Not every test needs a specialist firm. You probably do need one if any of these are true:

- This is your first pen test cited in a 510(k), De Novo, or PMA submission.



- Your device has hardware, wireless, cloud, and mobile components and your last test only covered one of them.
- Your launch date is fixed and a critical finding would slip revenue.
- You need a report that FDA reviewers already recognize and accept without rework.

**Ready to talk through your roadmap?**

**[Book a Strategy Session →](#)**

*Source guide: [12-critical-findings-from-medical-device-penetration-tests](#). For the most current version, visit [bluegoatcyber.com/guides/12-critical-findings-from-medical-device-penetration-tests](#).*