CYBERSECURITY WORKSHOP

Secure-by-Design: Five Things Every MedTech Technical Team Should Be Doing Today

Presented by Christian Espinosa, Founder & CEO, Blue Goat Cyber

MedTech Innovator Asia Pacific

MEDTECH INNOVATOR ASIA PACIFIC

WHY THIS ROOM

Cybersecurity is now a diligence item, not just a compliance step.

You're being evaluated this week. Judges score technical readiness, and investors ask about security posture before term sheets get written. A missing threat model or SBOM reads as risk, not as “not built yet.”

- Investors increasingly ask about security posture during diligence, before terms
- Judges scoring technical maturity look for the same evidence a regulator would
- The five things in this session are what both groups are actually checking for

The rule: the same five things that satisfy FDA are what satisfies diligence.

WHY THIS SESSION

Someone is on the other end of what you're building.

Every connected device your team ships eventually reaches a patient, a nurse, a caregiver. The code is not abstract. It's load-bearing for someone's safety.

- The engineer who names the attack path, not just “it's encrypted.”
- The product manager who scopes security into the roadmap, not as a stretch goal.
- The QA lead who tests the control, not just checks the box.
- The RA lead who traces the claim to evidence.

The rule: cybersecurity is a patient safety discipline, not a compliance checkbox.

WHO WE ARE

275+ submissions supported. Medical device cybersecurity only.

Blue Goat Cyber is an SDVOSB specializing exclusively in medical device cybersecurity: threat modeling, security risk management, SBOM, architecture views, and regulatory submission support across FDA and international frameworks.

275+

FDA PREMARKET SUBMISSIONS SUPPORTED

FRAMEWORKS WE WORK ACROSS

- **FDA**
- **EU MDR**
- **PMDA**
- **TGA**
- **NMPA**

The same underlying practices apply no matter which regulator you're targeting: threat modeling, SBOM, exploitability scoring.

Bottom line: this is what we've seen work, and where we've seen submissions stall.

THE SCOPE CHECK

Three questions decide if 524B applies to your device.

Section 524B(c) defines a “cyber device” as one that meets all three. Miss one, and 524B doesn't apply, at least not yet.

- Includes software, including firmware or programmable logic
- Can connect to the internet, even briefly or unintentionally: Wi-Fi, Bluetooth, cellular, USB, ethernet, and serial ports all count
- Contains characteristics that could be vulnerable to cybersecurity threats

The rule: if the first two hold, the third almost always does. That's not a loophole, that's the definition.

THE REGULATORY REALITY

Cybersecurity is no longer a gap any major regulator tolerates.

FDA's Section 524B (effective March 29, 2023) is the most detailed example, but it isn't the only one. Every major market now expects the same thing.

- FDA (US): Section 524B cybersecurity information requirement
- EU MDR: Annex I General Safety and Performance Requirements
- PMDA (Japan): guidance aligned with IMDRF international standards
- TGA (Australia) / NMPA (China): equivalent premarket cybersecurity expectations
- PMDA often accepts FDA cybersecurity documentation as evidence; MDSAP lets one audit satisfy five markets

The rule: build to FDA's bar once. PMDA often accepts that documentation directly, and MDSAP covers five markets in one audit.

Five things, one thread.

1. Threat model first
2. Control every interface
3. SBOM as living inventory
4. Score exploitability, not intuition
5. Design for postmarket

Every one of the five traces: threat → control → test → patient harm.

PRACTICE 1 · THREAT MODELING

Threat modeling comes before the architecture diagram, not after.

Teams that threat model at the end are documenting decisions already made. Teams that threat model first are making better decisions.

- Data flow diagram drawn first
- Threats identified per element
- Controls designed in, not bolted on
- Architecture reflects the controls

The rule: if the threat model can't get changed the architecture, it was done too late.

THE METHOD

DFD3 draws the boundaries. STRIDE classifies. A rubric scores.

These are three separate steps, often collapsed into one. That's where teams lose rigor.

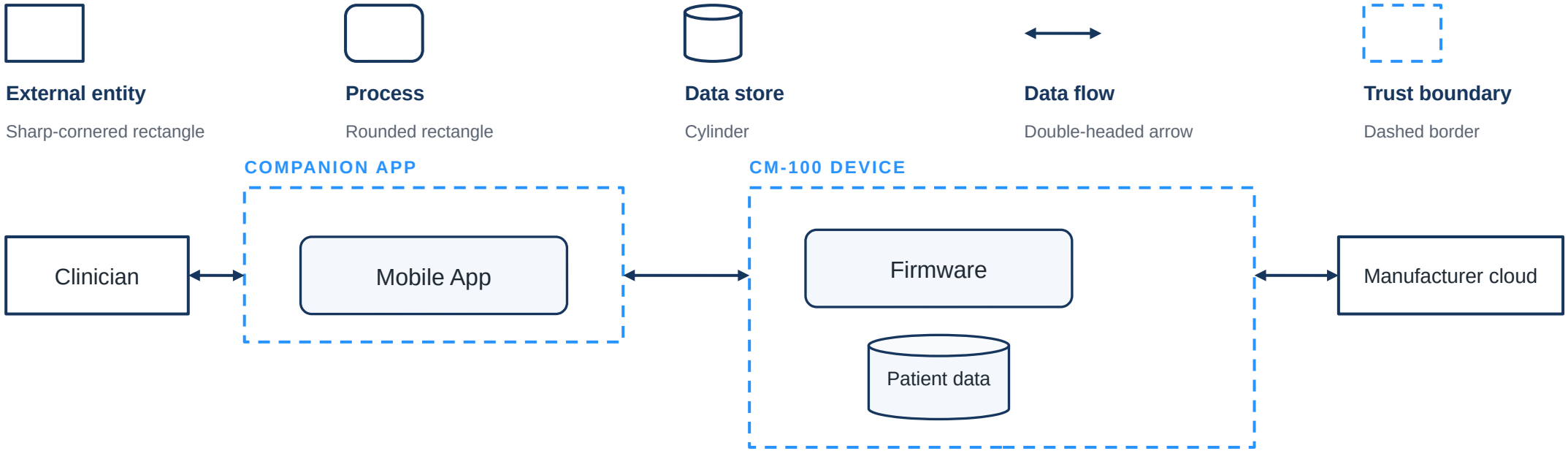
Step	What It Does	Output
DFD3 (Shostack)	Maps data flows and trust boundaries	A diagram, not a narrative
STRIDE	Classifies each threat by type, per element	A labeled list of threats
Rubric	Scores exploitability and harm	A prioritized action list

The rule: STRIDE tells you what kind of threat it is. It doesn't tell you how bad. That's a separate step.

WORKED EXAMPLE

A DFD3 example, on the Northwind CM-100.

Five icons, drawn on our running example. This is the diagram a threat model starts from.



Read it left to right: the dashed lines are trust boundaries. Every crossing is a place STRIDE gets applied.

PRACTICE 1 · THREAT MODELING

Threat modeling, done right.

NOT THIS

“We did a threat model for the submission.”

A document produced after the design already froze, describing decisions rather than shaping them.

THIS

A threat model that changed where authentication sits in the architecture, three months before code freeze.

PRACTICE 2 · ACCESS CONTROL

Segmentation is not authentication.

A device on an isolated VLAN is not a secured device. Segmentation reduces exposure. It doesn't verify who or what is on the other end of a connection.

- Every interface needs its own control
- Network segmentation is not a substitute for any of them
- “It's internal” is not an exemption

The rule: every interface gets a control, authenticated, restricted, or disabled. None of them get a pass because “it's internal.”

THE BAR

“Every interface” is longer than the list most teams write down.

Reviewers look for a named control per interface, not one asserted once for “the device.”

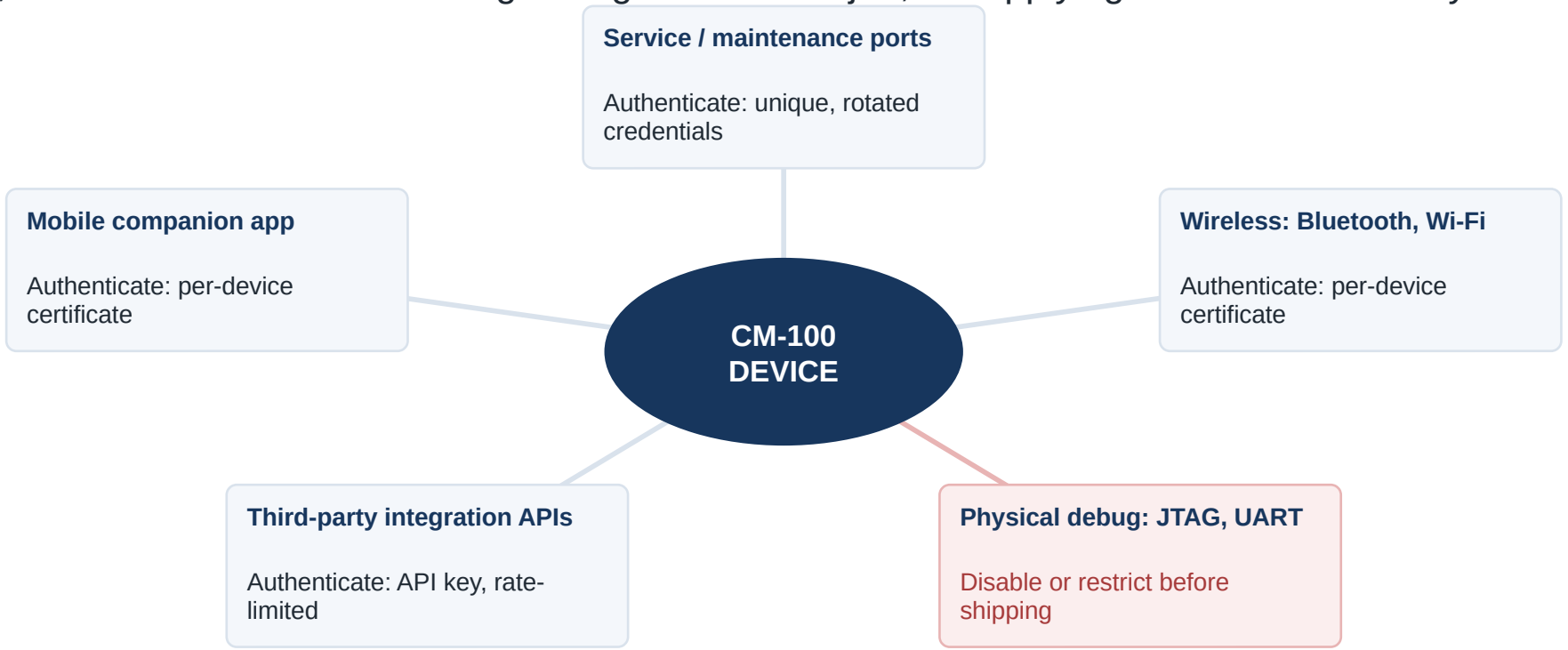
- Service and maintenance ports
- Wireless: Bluetooth, Wi-Fi
- Physical debug: JTAG, UART. Disabled or access-restricted before shipping
- Third-party integration APIs
- Mobile companion apps

Name it: which interface, which control, which credential, not “the device is secured.”

PRACTICE 2 · ACCESS CONTROL

Same interface list. Different control per type.

Five interfaces, five different controls. Naming the right one is the job, not applying the same one everywhere.



The rule: authenticate where you can verify identity. Disable where you can't.

PRACTICE 2 · ACCESS CONTROL

Access control, done right.

NOT THIS

A hardcoded default password on the maintenance port, justified because “it's not internet-facing.”

THIS

Per-interface, unique-per-device credentials with a documented rotation and lockout policy.

PRACTICE 3 · SBOM

Your SBOM is inventory, not paperwork.

An SBOM produced once, for the submission, is already stale the day it's filed. A living SBOM is a component inventory your team actually monitors.

- Required under 524B(b)(3) in the US, and under equivalent component-transparency rules for EU MDR, PMDA, TGA, and NMPA
- Tied to a vulnerability monitoring feed, not a one-time export

The rule: if your SBOM isn't tied to a monitoring feed, it's a snapshot, not an inventory.

THE BAR

What a reviewer actually checks in an SBOM.

FDA's expectation and ANSI/AAMI SW96 §10.2.2 both point to the same thing: components named, versioned, and monitored, plus two fields teams routinely skip.

- Component name and version
- Known vulnerabilities at time of filing
- Level of Support (LOS): actively maintained, no longer maintained, or abandoned
- End-of-Support (EOS) date, per component

The rule: a component without an LOS and EOS date isn't finished. FDA asks for both, per component.

WORKED EXAMPLE

A filled-out SBOM row, on the Northwind CM-100.

This is what “component name and version” from the last slide actually looks like: five real rows, two of them failing.

Component	Version	LOS	EOS	Known Vulnerabilities
Zephyr RTOS	3.6.0	Actively maintained	2027-06	None open
BlueZ (Bluetooth stack)	5.68	Actively maintained	2028-01	1, patched
OpenSSL	3.0.13	Actively maintained	2026-09	0 open
Qt UI Framework	5.15.2	**No longer maintained**	**2025-05 (past)**	**2 open, unpatched**
Legacy JSON parser	1.2.0	**Abandoned**	**Unknown**	**1 open, no fix available**

The rule: “abandoned” and an unknown EOS date are the two answers that trigger a deficiency.

PRACTICE 3 · SBOM

SBOM, done right.

NOT THIS

An SBOM generated once at submission time, then filed away.

THIS

An SBOM wired to a vulnerability feed, reviewed on a defined cadence, feeding the postmarket risk process.

PRACTICE 4 · EXPLOITABILITY

Exploitability, not probability.

“How likely is someone to try this?” is the wrong question. “Can this actually be reached and exploited, given the real attack path?” is the right one.

- CVSS v4.0 scores exploitability characteristics, not intent, not motive
- The FIRST calculator is the authoritative scoring source

The rule: score the path, not your guess about the attacker.

THE TOOL

The FIRST calculator turns metrics into a rating.

CVSS v4.0's base metrics determine how exploitable a vulnerability is. The calculator combines them into one rating, you don't compute it by hand.

CVSS v4.0 BASE METRICS

- Attack Vector
- Attack Complexity
- Attack Requirements
- Privileges Required
- User Interaction

EXPLOITABILITY RATING

- High
- Medium
- Low

The rule: score it at first.org/cvss/calculator/4.0. Read the exploitability rating, don't compute it by hand.

THE METHOD

Exploitability by severity decides controlled or uncontrolled.

FDA's postmarket guidance pairs exploitability against severity of patient harm to make one binary call: controlled, or uncontrolled. Illustrative pattern below, consistent with that approach.

	SEVERITY OF PATIENT HARM				
EXPLOITABILITY	NEGLIGIBLE	MINOR	SERIOUS	CRITICAL	CATASTROPHIC
High	Controlled	Controlled	Uncontrolled	Uncontrolled	Uncontrolled
Medium	Controlled	Controlled	Controlled	Uncontrolled	Uncontrolled
Low	Controlled	Controlled	Controlled	Controlled	Uncontrolled

The rule: there's no cell where catastrophic harm is called controlled. Remediate uncontrolled risk before release.

THE METHOD

Naming the attack path before scoring it.

“We encrypted it” is a claim. A named protocol, algorithm, and configuration is evidence.

NOT THIS

“The channel is encrypted.”

THIS

“AES-256-GCM channel encryption; certificate-based mutual authentication; attack requires network access plus a valid device certificate.”

The rule: name the protocol, the algorithm, the configuration, or it isn't scoreable.

PRACTICE 5 · POSTMARKET

Ship with a patch path already built.

A device without a designed update mechanism can't meet its own postmarket obligations, no matter which regulator you're submitting to.

- 524B(b)(2)(A) [US]: updates for known unacceptable vulnerabilities on a reasonably justified regular cycle
- 524B(b)(2)(B) [US]: out-of-cycle updates as soon as possible for critical vulnerabilities that could cause uncontrolled risk
- EU MDR, PMDA, TGA, and NMPA all require equivalent postmarket vigilance and reporting

The rule: “regular cycle” and “as soon as possible” are two different response times. Design for both.

THE BAR

Disclosure and KEV monitoring are architecture, not response.

A patch mechanism designed after a vulnerability is reported is a fire drill. One designed into the device from the start is a capability.

- A coordinated vulnerability disclosure process (SW96 §10.2.2) for external researcher reports
- Active monitoring against the CISA Known Exploited Vulnerabilities catalog
- An update mechanism that can actually deliver the patch to fielded devices

The rule: if the update mechanism is still a whiteboard idea, the device isn't done.

PRACTICE 5 · POSTMARKET

Postmarket, done right.

NOT THIS

No documented process for how a security researcher reports a finding.

THIS

A published, monitored coordinated disclosure channel with a defined acknowledgment and remediation timeline.

WORKED EXAMPLE

All five, traced on the Northwind CM-100.

One trust boundary, the wireless link between the CM-100 and its base station, walked through all five practices.

Threat	STRIDE: spoofing, unauthenticated pairing on the wireless link
Control	Per-device certificate authentication on the wireless interface, named and configured
Exploitability	Network-reachable, but requires a valid device certificate. Scored, not guessed.
Test	Penetration test case verifies pairing rejects an unauthenticated device
Patient harm	Unauthorized command injection to a cardiac monitoring device, mapped in the ISO 14971 risk file

Read it left to right: this is the spine. Every one of the five things is a link in this chain.

THE DEFICIENCIES WE SEE

The five gaps we catch before submission.

Threat model finished after architecture froze	→ **Move it to design kickoff**
Segmentation used as the only access control	→ **Control every interface individually, named by type**
SBOM treated as a one-time snapshot	→ **Wire it to a monitoring feed**
“Encrypted” / “validated” claims with no protocol named	→ **Name the algorithm, version, configuration**
No documented patch or disclosure path	→ **Build it before submission, not after an incident**

The rule: these aren't submission problems. They're design problems that show up at submission.

WHAT TO TAKE AWAY

Five lines, no notes needed.

- Threat model before you architect.
- Control every interface: authenticate, restrict, or disable. Segmentation is not a substitute.
- Treat the SBOM as living inventory.
- Score exploitability, not intuition, and name the attack path.
- Design the patch and disclosure path before you ship.

CLOSE

Compliance is the floor, not the ceiling.

These five practices don't just satisfy a reviewer. They build a device your team can defend, patch, and stand behind for its full lifecycle.

Before you leave: run the three-question scope check on one of your devices.

Slides & takeaways: bluegoatcyber.com/talks · Questions: Christian Espinosa

MEDTECH INNOVATOR ASIA PACIFIC