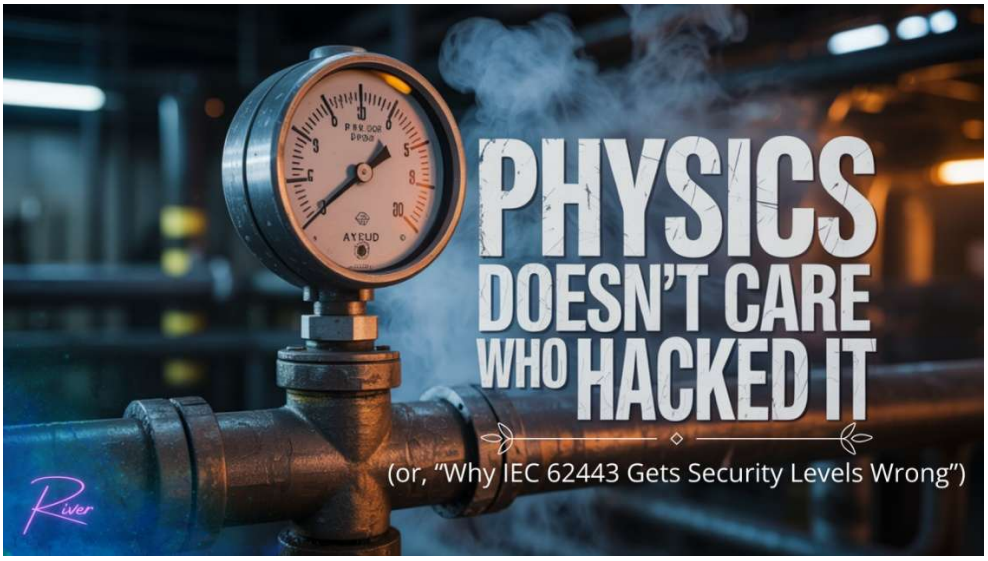
IEC 62443 Gets Security Levels Wrong, and Here's Why

River Caudle, November 26, 2025



"The way the standard defines Security Levels is backwards."

The standard that's supposed to protect your plant has a fundamental flaw baked into its foundation.

I've spent 20+ years in networking, most of them across industrial facilities - from nuclear plants to offshore platforms to food processing lines. I've implemented IEC 62443 standards more times than I can count.

And I'm here to tell you: **the way the standard defines Security Levels is backwards.**

The Problem: We're Profiling Attackers, Not Protecting Plants

Open IEC 62443-1-1 and look at how Security Levels are defined. The standard categorizes threats by *who* is attacking you:

- **SL-1:** Casual or coincidental violation
- **SL-2:** Intentional violation using simple means (hackers)
- **SL-3:** Sophisticated attack with moderate resources (hacktivists, organized crime)
- **SL-4:** State-sponsored attack with extensive resources (nation-states)

See the assumption buried in there? The standard implies you can *predict* who's coming for you. That somehow the sophistication of your adversary correlates neatly with the severity of what happens when they win.

This is academically elegant and operationally useless.

Physics Doesn't Care About Attribution

Here's what the standard gets wrong: **The boiler doesn't care who hacked it.**

If a command opens a relief valve when it shouldn't, the pressure vessel fails. The explosion that follows doesn't pause to check whether the attack originated from a Russian GRU officer or a bored teenager running a script they found on GitHub.

The consequence is identical. The funerals are identical. The crater is identical.

Yet the standard would have you believe that defending against the "script kiddie" (SL-2) somehow requires less rigor than defending against the "nation-state actor" (SL-4). As if the laws of thermodynamics adjust themselves based on the attacker's budget.

The Real-World Mismatch

Consider this scenario: You have an unpatched safety system controlling a high-pressure process. The vulnerability is trivial to exploit - a known CVE with public proof-of-concept code.

According to IEC 62443's attacker-based model, a script kiddie using that exploit is an "SL-2 threat." But if they succeed?

You get an SL-4 funeral.

The mismatch is obvious once you see it. We're categorizing inputs (attacker capability) when we should be categorizing outputs (operational consequence).

The "Risk Assessment" Defense

Before the ISA sends their hitmen, I know what the purists will say: "The standard covers this in the Risk Assessment (RA) phase! You calculate Risk as Likelihood x Consequence, so high-consequence assets get high security."

Theoretically? Yes. But practically? The Detailed Risk Assessment (ZCR 3.2) fails in the field because it over-complicates the "Likelihood" variable. By tying the Security Level definitions to the attacker, the standard forces engineers to play counter-intelligence analyst.

Here is the reality: In a connected world, Likelihood should be assumed to be 1.

If a system is accessible, it will be probed. Therefore, the Security Level shouldn't be a gamble on *if* it happens; it should act like a Safety Integrity Level (SIL)… a rating based purely on the unmitigated consequence of failure.

A Better Framework: Consequence-Based Security Levels

In operations, we don't actually care *who* attacks us. We care *what happens when they win*.

Here's how Security Levels should actually work:

SL	The Consequence	Operational Reality
SL-1	Nothing / Mild Inconvenience	Nuisance alarm. Someone investigates. Coffee gets cold.
SL-2	Production Affected	Line slows, quality suffers, scrap increases. Money leaks.
SL-3	Production Halted / Equipment Damaged	Full stop. Metal is bent. Capital expenditure required.
SL-4	Hazard or Death Likely	Explosion, release, environmental catastrophe. Funerals.

Reality-based security levels

This fundamentally changes how you classify systems and allocate resources.

The escalation thresholds become observable questions any operator can answer in 30 seconds:

- **SL-1 → SL-2:** Did it cost us money?
- **SL-2 → SL-3:** Did it stop us or break something?
- **SL-3 → SL-4:** Can it hurt someone?

No threat intelligence required. No guessing whether your adversary has a government budget. **Consequence-based classification is observable and immediate.** Attacker-based classification requires crystal balls and security clearances.

Why This Matters for Your Budget

When you define Security Levels by attacker capability, you end up in absurd conversations with leadership: "We need to defend against nation-states!"

Leadership hears: "We need to outspend China." They check the budget and conclude this is impossible.

So they do… less.

When you define Security Levels by consequence, the conversation changes: "This system, if compromised by *anyone*, can kill people. It requires SL-4 controls."

Now you're not trying to predict adversaries. You're engineering against failure modes. That's something that operations people understand. That's something we've been doing since the first pressure vessel was forged.

The Practical Application

Walk your facility with this lens:

1. Identify what each system controls
2. Ask: "What's the worst thing that happens if an adversary gains full control?"
3. Assign the SL based on that consequence - not on your guess about who might attack

A system controlling a critical relief valve gets SL-4 treatment regardless of whether you think nation-states are interested in your plant. Because the 17-year-old who stumbles into your network doesn't need nation-state resources to cause nation-state consequences if your architecture lets them reach that valve.

From Framework Flaw to Field Implementation

Recognizing this mismatch is step one. Actually, implementing consequence-based security architecture is where most organizations stall.

We need a bridge between the standard's intent and the operator's reality.

This is exactly why I developed the **SECURE Method** - a practical framework that translates IEC 62443's valuable concepts into operations-focused implementation. Consequence-based Security Level classification is built into the methodology from the ground up.

The SECURE Method covers:

- Consequence-based zone classification (not attacker profiling)
- Conduit design that operations teams can actually maintain
- Security requirements that align with operational priorities
- Implementation sequences that don't require IT permission slips

I teach this in hands-on workshops where we work through real architecture - not theoretical case studies. Small class sizes, direct engagement, and you leave with documentation you can actually use.

The Bottom Line

We don't spend money to stop "*Nation States*."

We spend money to stop "**Explosions**."

IEC 62443 is a valuable framework, but its attacker-based Security Level definitions were written by people who think about security as a discipline unto itself. Those of us who actually operate plants know that security isn't the goal - *operational continuity* is the goal, and *safety* is the constraint.

Define your Security Levels by consequence, not by enemy. Because the physics doesn't care who sent the command.

Author

CSO, River Risk Partners | Industrial Cybersecurity, Risk & Production Loss Prevention | Nuclear, Energy & Critical Infrastructure | Author & Strategist | riverrisk@proton.me