.experience

Cybersecurity
A modern Wild West tale

A magazine by ERNI since 1999.

Editorial

Dear Readers,

Cybersecurity is often discussed in terms of tools, incidents and regulations. Yet behind every secure or insecure system are people making decisions every day.

In this issue of .experience, we explore cybersecurity through a different lens, a tale of three characters: 'The Good, the Ugly and the Bad'. The Good builds software with security in mind from the very beginning. The Ugly ensures compliance, governance and accountability in a growing regulatory landscape. The Bad challenges assumptions by actively trying to break systems before real attackers do.

These perspectives come to life through three reference cases featured in this issue. You will read about how security best practices can be exported into professional development teams, how security can be designed across the entire product lifecycle, and how penetration testing reveals real-world risks that often remain hidden.

We invite you to join an insightful exploration and find inspiration on how to apply these insights in your own teams.

Sincerely,

Pavo Kohler
CEO, ERNI Group

Table of contents

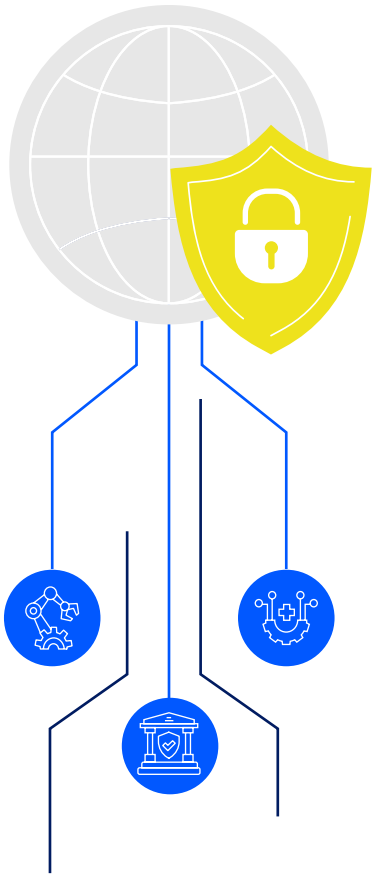
Cybersecurity in times of uncertainty

In today's interconnected world, cybersecurity is no longer an IT issue. It is a business imperative. Organisations face a landscape shaped by rapid digitalisation, geopolitical tensions, economic uncertainty and constantly evolving threats. Software powers critical services, data drives decisions, and every digital connection represents both opportunity and risk.

By Albert Alsina, Managing Director, ERNI Spain

Impact on specific domains

The scope of cybersecurity has expanded far beyond protecting networks and devices. In financial services, attacks on payment systems or sensitive data can have immediate global consequences. In healthcare and life sciences, breaches can put patients at risk. Manufacturing and critical infrastructure face threats that could disrupt operations or even safety. Across sectors, attackers exploit complexity, speed and human error, and the stakes have never been higher.

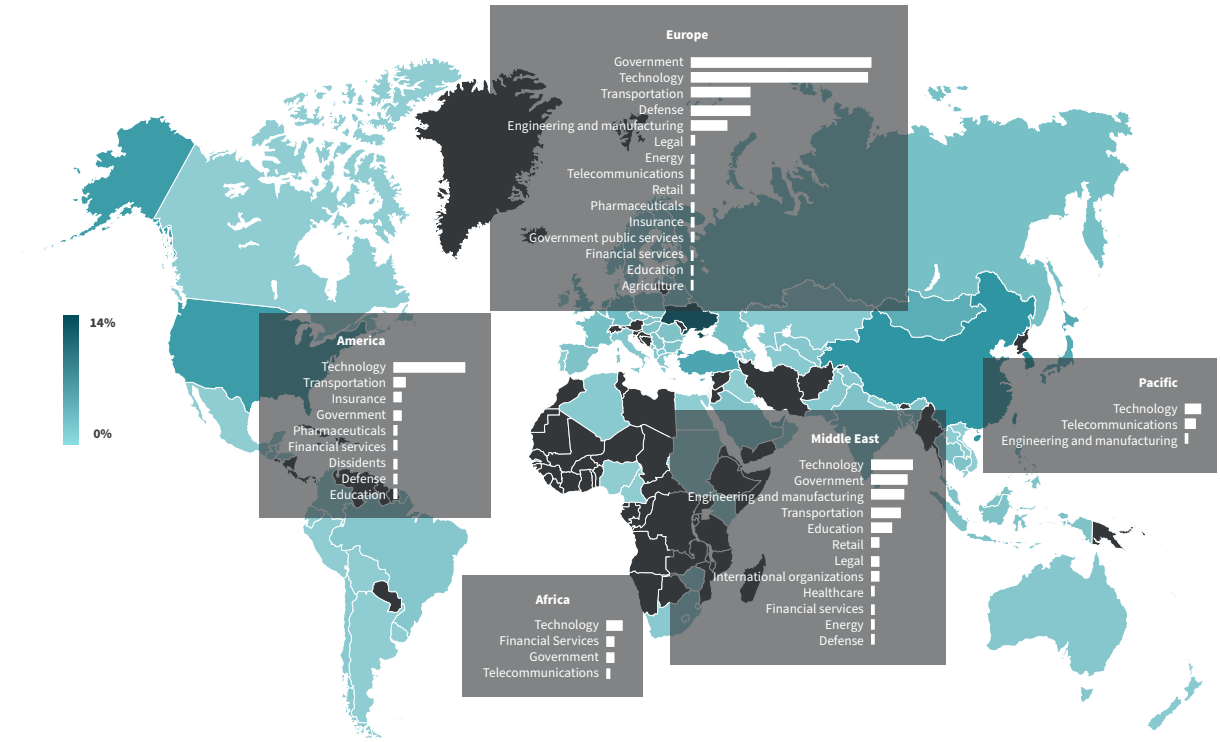


According to the ENISA Threat Landscape report 2025, in Europe, the financial sector is the third most targeted domain, with 46% of attacks impacting European banks (credit institutions), followed by 13% of attacks affecting public financial organisations.

The uncertainty of today's environment amplifies these risks. Geopolitical tensions can trigger state-sponsored attacks. The methods are becoming more sophisticated, creating a landscape that is more volatile and treacherous than ever before.

The World Economic Forum reports that *72% of global executives now take geopolitical events into account in their cybersecurity strategies.* Also, economic pressures drive organisations to accelerate digital projects, sometimes at the expense of thorough security practices. Supply chain dependencies and cloud adoption introduce additional layers of vulnerability. And while technology evolves, attackers continuously adapt, finding new entry points in AI, IoT or remote working environments.

The most targeted sectors worldwide



Source: ESET - APT Report 2024-2025

Cybersecurity as a strategic capability

In this context, cybersecurity must be approached as a strategic capability, not a reactive measure. Proactive security in software development, thorough testing of applications, and compliance with emerging regulations like the Cyber Resilience Act are no longer optional – they are essential to resilience and trust. Organisations that embed security across processes, invest in defensive and offensive capabilities and adopt a forward-looking perspective can navigate uncertainty with confidence.

Connected systems, real-world consequences

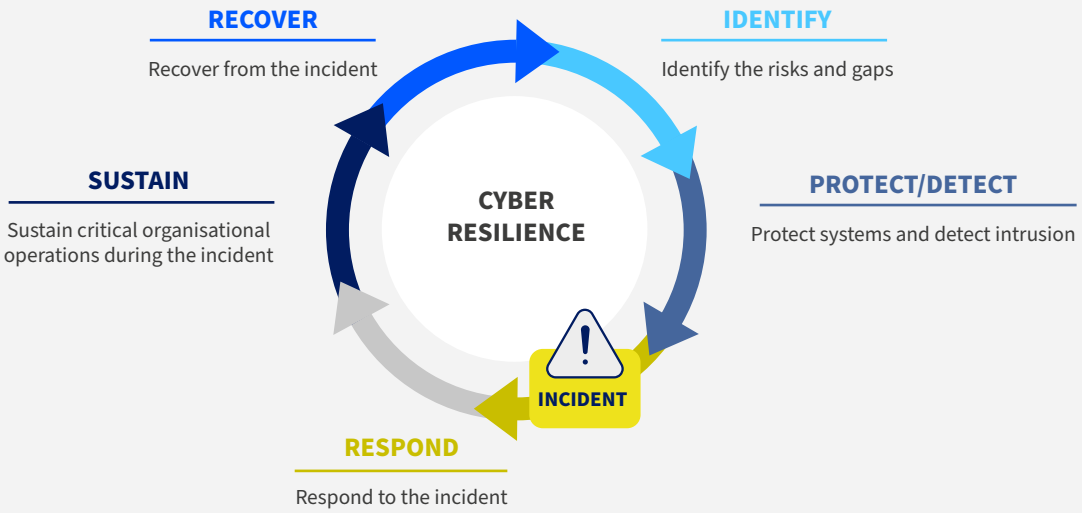
Today's products rarely operate in isolation. They are part of interconnected ecosystems that directly affect everyday life. Smart homes rely on secure interfaces to protect privacy and comfort. Medical devices must communicate reliably to safeguard patient safety. Industrial systems depend on secure automation to prevent downtime and production losses. Vehicles exchange data continuously to enable real-time safety on the road.

In all these domains, trust is assumed until it is broken. When security fails, the impact is immediate and tangible. Regulations such as the Cyber Resilience Act aim to make security visible, measurable and maintainable throughout a product's lifecycle, shifting the focus from isolated controls to systemic resilience.

What resilience looks like in practice

Resilient systems do not emerge by chance. They are the result of deliberate design choices and disciplined execution. Security must be considered from the earliest stages, through threat modelling, clearly defined security requirements and alignment with relevant standards. Decisions made during design often determine whether security later becomes an enabler or an obstacle. Equally important is integrating security into everyday development workflows.

5 stages of cyber resilience



Source: Weforum.org/stories/2022/07/how-do-you-safeguard-a-city-from-cyber-attacks/

Secure-by-default architectures, software bills of materials and CI-integrated vulnerability scanning allow teams to address risks continuously rather than reactively. Security becomes part of how software is built, not something added at the end.

Resilience also extends beyond release. Secure deployment, observability, patch management and monitoring are essential to maintain operational integrity over time. Continuous testing through penetration testing, red teaming and proactive assessments ensures that risks remain visible, manageable and reduced as systems evolve.

Conclusion

This issue of .experience explores the full spectrum of cybersecurity challenges and responses. You will learn how secure development practices prevent vulnerabilities, how regulatory frameworks are shaping responsibilities, and how offensive testing provides insight into real-world risks, as told by a tale of three characters – 'The Good, the Ugly and the Bad' – representing developers, compliance officers and ethical hackers.

By understanding the broader context, organisations can turn uncertainty into opportunity and build the foundations for long-term resilience. Cybersecurity is no longer a matter of 'if' but 'how'. The organisations that succeed will be those that treat security as an integral part of their strategy, aligning technology, processes and people to the realities of a fast-changing world.

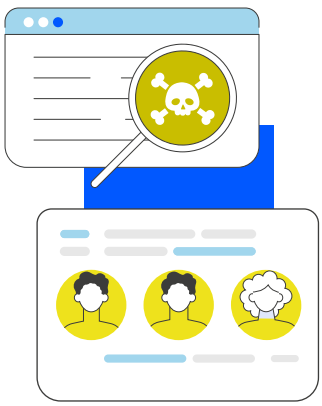
About Albert Alsina

Albert Alsina is the Managing Director of ERNI Spain, bringing 10 years of ERNI experience across project leadership, client management and talent development. He leads nearly 300 employees, fostering growth, innovation and impactful projects. Passionate about empowering teams, Albert balances work with family, sport, music and cooking.

‘The Good, the Ugly and the Bad’: A modern cybersecurity fable

Cybersecurity is a shared responsibility in a world full of modern complexities and new threats. To keep systems secure, three roles work together: secure software developers, compliance specialists and ethical hackers. In this article, we tell a fictional yet strikingly familiar story about ‘The Good, the Ugly and the Bad’ – with their strengths and essential characteristics in a security-minded organisation.

By David Soto Dalmau, Cybersecurity Principal, ERNI Spain



There’s something timeless about a classic Western. A lone rider crosses a barren landscape, danger lurking behind every rock, trust as scarce as water in the desert. Guns are drawn, deals are broken and survival depends not just on who’s the fastest – but on who understands the game.

Swap the revolvers for laptops, and the dusty towns for networks and systems, and suddenly the Wild West doesn’t feel so distant.

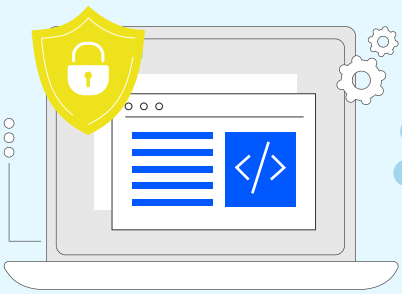
In cybersecurity, much like in the old Westerns, we navigate an unpredictable frontier. Threats come unannounced. Allies may turn. And victory isn’t about brute force – it’s about outsmarting your adversary, securing your turf and knowing when to shoot... And when to code.

At ERNI, we recognise these dynamics all too well. Every client engagement is a new town, with its own rules, its own sheriff and its own lurking dangers. That’s why we ride into each project with a trio of experts: ‘The Good, the Ugly and the Bad’. Each one with a purpose, each one with a role, and none of them truly effective without the others. This isn’t just a metaphor – it’s a method. And it’s what guides our approach to building, regulating and testing secure systems in a digital world that often feels as lawless as the frontier.

And it is at this frontier where our tale begins...

In a far-off digital Wild West, where firewalls are the new adobe walls and bounty hunters are called ‘pentesters’ (penetration testers), three figures ride into the cybersecurity frontier: ‘The Good, the Ugly and the Bad’. This isn’t a tale of shootouts at noon, but of simulated attacks, tedious regulations and unsung heroes who code with principles. A paradox: to protect ourselves from chaos, we need a little chaos, a dash of law and a guardian who knows when to draw their trusty weapon: code.

The Good:
Secure development



The Good wears a digital lab coat, fingers stained with coffee and clean code. They're the ones who design securely from the start, applying 'secure by design' principles and reviewing every component, every dependency and every endpoint.

Their ethics are grounded in professionalism – not fear of auditors or attackers, but respect for users.

The Good documents, validates input, encrypts communications and anticipates what might go wrong before anything does. They rely on models like STRIDE, use SAST and DAST tools, and follow frameworks like OWASP ASVS.

But they also have the soul of a storyteller: telling a tale of prevention, intelligent design and accountability.

The Ugly:
The regulations no one wants

The Ugly arrives in a grey suit with a clipboard. They talk about GDPR, ISO 27001, NIS2 and other abbreviations that make tech teams sweat. They're the auditor, the compliance officer, the risk manager. No one invites them to creative meetings, but everyone calls when something breaks.

Their ugliness lies not in function, but in perception: seen as a burden, not a shield.

But the Ugly embodies the Greek logos – rationality that imposes structures, controls and processes. They demand encryption evidence, access controls and vulnerability management. Yes, sometimes it feels like bureaucracy, but without them there are no boundaries, no accountability, no justice when systems fail. The Ugly turns "we should" into "we must" – unpopular, yet vital.

The Bad:
The ethical hacker

The Bad slips in the back door with a crooked grin. They wear hoodies, work in dark terminals, and ask uncomfortable questions. They're the pentester, the red teamer, the offensive cybersecurity consultant. Their role: to attack their own client.

From the outside, they seem villainous, but their ethics are clear: find flaws before the real villains do.

The Bad channels pathos – the urgency, the imminent risk, the adrenaline that reminds us no system is unbreakable.

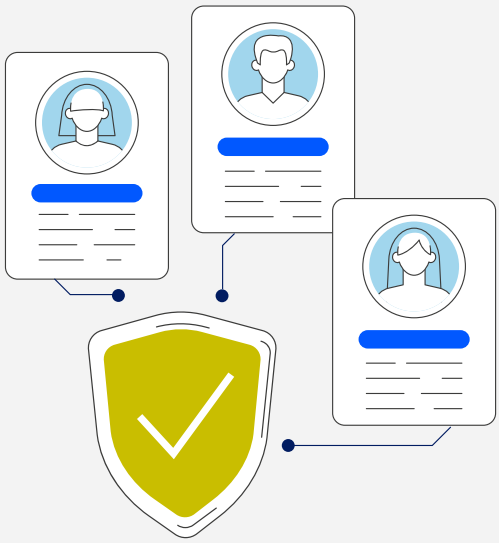
They wield Metasploit, Burp Suite, custom scripts and a lot of creativity. They break things so others can fix them. Though they sometimes annoy the Good or irritate the Ugly, they are essential to completing the security cycle.

A circular paradox

The Good builds secure systems but can't foresee everything. The Bad tests them, hunting for that 1% of human or technical error. The Ugly watches, documents, enforces and ensures that lessons aren't forgotten.

But it's also true that without the Good, the Bad would only find ruins; without the Bad, the Good would live in false confidence; and without the Ugly, all they'd learn would be lost in the chaos of sprints.

The paradox is that they need each other, challenge each other, complete each other. There is no security without design, validation and compliance. And no progress without accepting that the Bad may be right, and the Ugly – though annoying – protects what the good builds.



The ERNI approach: When the three ride together

At ERNI, we don't just tell this story; we live it. Our approach to cybersecurity integrates these three archetypes into our service model, aligning them with the real-world needs of our clients.

When our clients need robust, future-proof digital products, we bring in the Good: our secure software development teams, who apply best practices, from code reviews to threat modelling, ensuring that security is embedded from day one.

When compliance becomes critical, whether for healthcare, finance or public infrastructure, the Ugly steps in: our regulatory and governance experts guide organisations through the maze of legal and industrial requirements. They don't just tick boxes; they design systems that are auditable, reliable and resilient.

And when clients want to challenge their assumptions and harden their defences, the Bad gets to work: our offensive security teams perform ethical hacking, red teaming, and simulate real-world threats to test limits and expose blind spots.

Each role is valuable on its own, but at ERNI, we understand their greatest power comes from their synergy. We tailor this trio to the maturity, needs and industry of each client – sometimes starting with the Ugly to build foundational trust, other times unleashing the Bad to map attack surfaces, or letting the Good lead a greenfield product with security baked in.

We believe that security isn't just a feature – it's a practice. And by embracing all three perspectives, we help organisations not only defend themselves but grow with confidence in a digital world that changes faster than a gunslinger's draw.

Epilogue: A shootout avoided

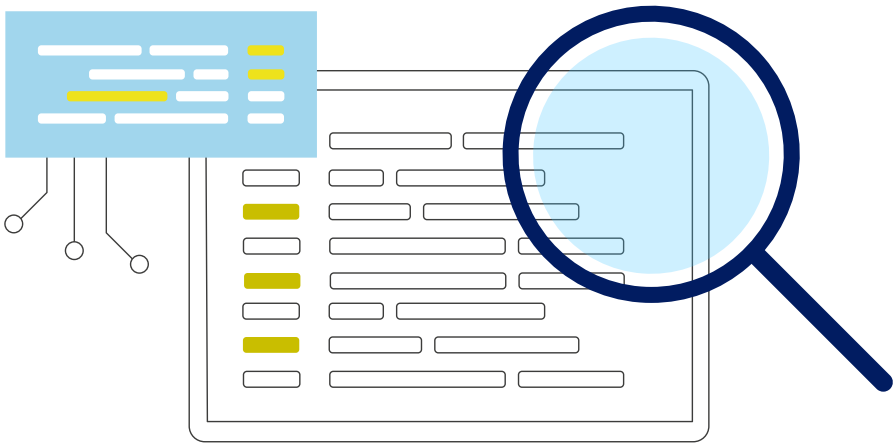
In this story, there's no final duel. The Good doesn't shoot the Bad, nor does the Ugly slap on handcuffs. Instead, they work together: the Good codes with foresight, the Bad tests with cunning, and the Ugly writes the rules of the game.

The moral is simple: effective cybersecurity isn't achieved by eliminating the Bad or ignoring the Ugly. It's achieved when all three understand that the digital frontier can only thrive if they ride together.

So, the next time you think about security, ask yourself: Where are your three outlaws? Because if one is missing, someone else might draw first.

Next episode: Into the frontier

This was just the opening scene. In the next articles, we'll ride deeper into the territory of each of these three figures – exploring their tools, their mindset and their code of honour. We'll see how the Good builds, how the Ugly governs and how the Bad breaks, all through real-world cases that bring their roles to life. Saddle up. The story's just getting started.



About David Soto Dalmau

David Soto Dalmau is Cybersecurity Practice Area Lead at ERNI in Barcelona. As a lifelong ethical hacking and CTF enthusiast, he combines expertise in cryptography, penetration testing and risk management with teaching and fostering a culture of continuous learning.

“I’m not a hero – I’m a secure software developer”

In a world of rapid digital development and constant cyber threats, The Good takes a different approach. Not the fastest coder or loudest voice, his work often keeps systems secure. In this exclusive interview, we explore what it truly means to build safe software today.

An interview with The Good (Samuel Hernández, Full-Stack Expert Consultant at ERNI Spain) on the digital frontier

.experience: How do you define your role in a development team?

The Good: I see myself as the one who ensures the foundation is solid before the house is even framed. My role is to think ahead – not just to deliver functionality, but to anticipate what might go wrong and build safeguards before code is ever executed. I work to make sure that what we ship is not just performant or scalable, but secure by design. It’s less about preventing attacks reactively and more about designing systems that are resilient from day one.

When does security begin in your process?

Before the first commit. Security starts with the architecture. Before we code, I want to know what data we’re handling, where it lives, who accesses it and what could go wrong. I use threat modelling frameworks like STRIDE to visualise potential vulnerabilities and build mitigations directly into the design. If security is an afterthought, it becomes technical debt. My job is to bake it in from the beginning.

Can you walk us through your daily toolkit?

My tools evolve depending on the project stage, but some staples remain. For early development, I define coding standards that avoid the most common pitfalls – no hardcoded credentials, no unchecked inputs, no insecure dependencies. In parallel, I implement static analysis tools (SAST) that scan code on every push. During integration, I add dynamic analysis (DAST) to test live applications for behavioural issues. I also rely on infrastructure-as-code validations and container scanning to make sure deployment environments are as trustworthy as the code itself.

I use dependency scanners linked to public vulnerability databases, and I maintain a detailed Software Bill of Materials (SBOM) for every build. Every package we use is accounted for, versioned and monitored. If a vulnerability is discovered in an upstream library, we already know exactly where it lives in our system.

What's your stance on secrets and credentials management?

Secrets should never be visible. No passwords in plain text, no API keys pushed to repositories. I integrate secrets management tools like Vault or cloud-native solutions to keep sensitive data encrypted and access-controlled. I apply the principle of least privilege rigorously – no user or process should have more access than strictly necessary. I also automate secrets rotation and monitor access logs. Secrets are treated like weapons: locked away, carefully tracked, and used only when necessary.

Beyond tools and code, what defines a secure developer?

Discipline. Secure development isn't about knowing every exploit – it's about being methodical. It's about making deliberate choices at every step: validating inputs, enforcing type safety, logging responsibly, thinking about failure modes. I also invest time in documentation, not just for my team today, but for whoever inherits the code tomorrow. Security thrives on clarity, and documentation is how we pass down intentions and warnings.

But more than that, it's about culture. I advocate for secure practices in code reviews. I host post-mortems that aren't about blame but about learning. I train teammates to spot red flags. My goal is not to be the only one thinking about security – it's to make sure everyone does.

How do you handle legacy systems or insecure inheritances?

With patience and a clear strategy. I start by mapping the system and identifying high-risk areas. From there, I conduct triage: what needs fixing now, what can be mitigated, and what should be deprecated. I focus on

isolation, input validation, and patching what we can, while lobbying for long-term reengineering when necessary. It's never ideal, but legacy systems are a reality. The key is to reduce their attack surface incrementally without paralysing the product roadmap.

What does success look like for someone in your role?

Ironically, success often means silence. No breach. No urgent hotfixes. No headlines. It means systems that just work – and keep working – without users ever realising the countless things that could have gone wrong but didn't. It also means knowing that if something does go wrong, we have logs, alerts and rollback mechanisms in place. Success is invisible until it's not.

Any misconceptions about secure development that you often encounter?

Yes, plenty. The biggest one is that secure development is slow. It can feel that way at first – especially in teams accustomed to speed over structure – but long term, it's faster. You avoid firefighting. You reduce rework. You build trust with clients and regulators. Another myth is that security is someone else's job – usually an external team or an auditor. I reject that completely. Security is everyone's job, but developers are on the front line. We have the power to prevent, not just respond.

How do you balance security with product pressure and deadlines?

That's one of the hardest parts of the job. The pressure to deliver fast is real, and security often looks like friction to people outside the process. What I try to do is integrate security into existing workflows, so it doesn't become a blocker. Automated tests, pre-commit hooks, lightweight reviews – these go a long way. But I also work to build credibility. When teams see that I'm not just pointing out problems but helping prevent future chaos, they begin to value that input. The balance comes from partnership, not from enforcement.

Have you ever failed? And what did you learn from it?

Absolutely. I've missed things. We all do. Once, a third-party package I trusted introduced a critical vulnerability in a patch release. I had SBOMs, but I wasn't enforcing version locks strictly enough. We caught it fast, but it was a wake-up call. Since then, I've treated third-party code with even more scrutiny. Failure teaches you humility – and it teaches you to listen more to your systems, your alerts and your intuition.

What motivates you to keep doing this work?

I believe that technology has immense power, but with that comes immense risk. And too often, we build first and think later. I don't want to live in a world where security only applies after someone gets hurt. I want to build things that empower without endangering. That's what keeps me going. Knowing that quiet, thoughtful work today can protect someone else's future.

> The pressure to deliver fast is real, and security often looks like friction to people outside the process. What I try to do is integrate security into existing workflows, so it doesn't become a blocker.

What sets a cybersecurity champion apart from other developers?

A cybersecurity champion isn't just someone who knows how to write secure code – they're someone who brings that mindset into the team and advocates for it consistently. What sets them apart is not technical brilliance, but presence. They're the ones who ask uncomfortable questions in sprint planning, who raise their hand when a shortcut might become a liability. They don't just fix vulnerabilities – they help prevent their recurrence. Champions are bridges: between developers and security specialists, between the business and the tech. They create a culture where security is part of the definition of done.

> A cybersecurity champion isn't just someone who knows how to write secure code – they're someone who brings that mindset into the team and advocates for it consistently.

Final thoughts?

I'm not a hero. I don't chase threats. I don't look for glory. I build for durability. My job is to make sure what we create can be trusted. Because in the end, trust is the most valuable currency in tech. And it's earned one secure line at a time.

About Samuel Hernández

Samuel Hernández, our Full-Stack Expert Consultant at ERNI Spain, develops solutions connecting medical devices with laboratory systems, using .NET, Angular, cloud technologies, and secure, compliant practices. Passionate about innovation and reliable software, he ensures high-quality, user-focused solutions in complex regulated environments.

A 360º approach: Security best practices for professional dev teams

Security is more than a checklist. This case shows how we exported a 360º approach to professional development teams, combining principles, best practices and CI/CD enforcement. Teams gained clarity, confidence and alignment, making security a natural part of software quality.

By David Soto Dalmau, Cybersecurity Principal, ERNI Spain

The challenge: Security knowledge without a unifying framework

In many organisations, software security initiatives grow organically. Teams receive guidance on data protection, coding standards, vulnerability management, and tooling – but often as isolated topics, disconnected from one another.

The result is familiar: developers know what tools to use and what rules to follow, yet struggle to understand how all security concerns fit together across the software lifecycle.

The core need behind this training initiative was therefore not to introduce more controls, but to establish a shared, end-to-end security model that development teams could apply consistently – from design decisions to runtime behaviour.

A 360° perspective: Security as a continuous system

From the outset, the training was designed around a single idea: Secure software is not achieved through a single practice or tool, but through the alignment of principles, behaviours and controls across the entire development lifecycle.

To make this tangible, we structured the programme as a progressive journey covering three complementary dimensions:

- Foundational security principles
- Secure development best practices
- Operational security controls integrated into CI/CD

Each layer reinforced the next, creating a coherent and repeatable security mindset.

Foundations: The CIA triad as the security baseline

The journey started with first principles. Rather than jumping directly into tooling, the training established a common baseline using the CIA triad – Confidentiality, Integrity and Availability. This was not treated as theory, but as a practical lens through which all later decisions would be evaluated:

- What data must remain confidential?
- Which operations require integrity guarantees?
- What availability assumptions does the system depend on?

By anchoring security discussions in CIA, teams gained a neutral, technology-agnostic way to reason about risk and impact, independent of implementation details.

This foundation proved essential when later discussing trade-offs, priorities and failure scenarios.

Secure development best practices: From principles to behaviour

Building on the CIA baseline, the second part of the training focused on secure development practices, treating security as a quality attribute of software – on par with performance or reliability. Key topics included:

- Input validation as an explicit trust boundary
- Secure handling of secrets and sensitive material
- Session management and authorisation invariants
- Zero Trust principles applied at code level

The emphasis was deliberately placed on developer behaviour, not checklists. Security was framed as a set of invariants that must hold true, regardless of refactoring, feature growth or architectural evolution.

This approach helped teams recognise that many real-world incidents are not caused by exotic attacks, but by regressions – controls that once worked but were unintentionally broken by later changes.

From best practices to proof: Testing and verification

Once secure coding practices were established, the training moved to an often misunderstood area: security testing versus vulnerability scanning. A clear distinction was introduced:

- **Security testing** proves that controls work under attacker behaviour
- **Vulnerability scanning** detects known risk indicators at scale

This distinction removed a common source of false confidence and set realistic expectations for each category of tool.

Security testing was presented as a way to validate behaviour – negative paths, misuse cases and failure modes – while scanning was positioned as an essential but complementary source of signal.

Tooling integration: Adapting security controls to the stack

Only after principles and practices were clear did the training introduce security tooling, explicitly adapted to the technology stack in use. Rather than promoting a fixed toolchain, the focus was on:

- **Why** a tool exists
- **Where** it fits in the lifecycle
- **When** it should influence delivery decisions

Static analysis, dependency scanning, SBOM generation, runtime testing, and configuration scanning were all mapped to concrete pipeline stages, emphasising consistency of intent even across heterogeneous technologies and languages.

A special focus was placed on non-negotiable risks – such as leaked secrets or critical misconfigurations – which invalidate all other security guarantees and must always result in immediate action.

Security enforced through the pipeline

The final part of the training tied everything together into a single, operational model: security enforced continuously through the delivery pipeline. From pull requests to pre-release environments, each stage was associated with:

- A specific type of security signal
- A clear decision (block, track or proceed)
- Explicit ownership

This pipeline-centric view transformed security from an abstract concern into a repeatable execution model that teams could visualise, discuss and evolve.

Outcome: Alignment, clarity and confidence

The holistic nature of the training – covering principles, practices and execution – proved to be its key strength. Development teams reported:

- A clearer understanding of why security controls exist
- Increased confidence in how to apply them
- Greater alignment between security expectations and engineering reality

Most importantly, security was no longer perceived as an external constraint, but as an integrated part of professional software engineering.

Conclusion

Exporting security best practices to professional development teams is not about transferring rules, tools or compliance requirements. It is about transferring a coherent security model that spans the entire lifecycle.

When organisations approach secure development from a 360° perspective – grounded in principles, reinforced by best practices and enforced through automation – security stops being an afterthought and becomes part of how quality software is built.

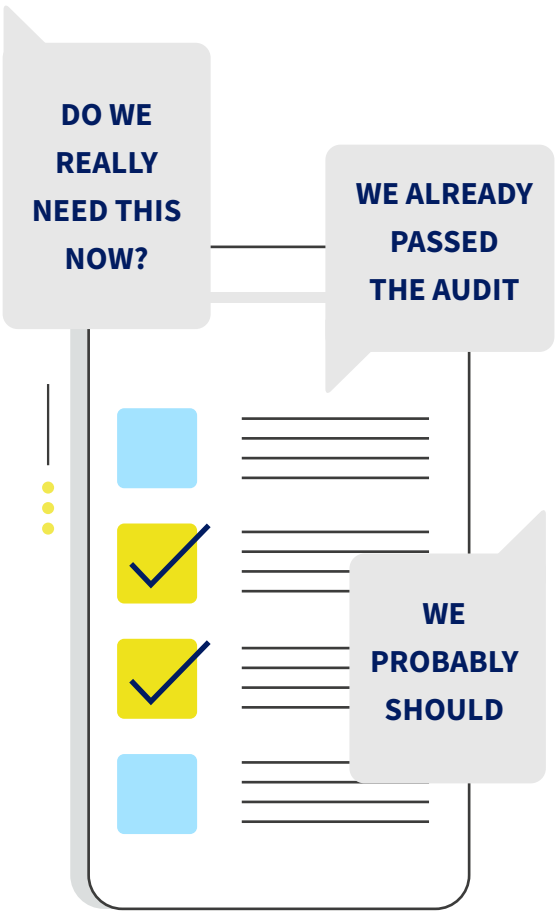
About David Soto Dalmau

David Soto Dalmau is Cybersecurity Practice Area Lead at ERNI in Barcelona. As a lifelong ethical hacking and CTF enthusiast, he combines expertise in cryptography, penetration testing and risk management with teaching and fostering a culture of continuous learning.

The Ugly‘s side: Why compliance isn‘t what you think

Compliance often plays the villain in cybersecurity: policies, audits and awkward questions nobody enjoys. But behind the checklists is experience that was earned the hard way. This article gives the Ugly a voice and shows why compliance is not about control, but about protecting people, products and trust.

By José Francisco Agulló, Quality Manager, ERNI Spain



Yes, I’m the Ugly – And I want you to understand why

You know the type. The person with the checklist, talking ISO 27001, NIS2, GDPR, CRA and more. The compliance officer sidestepped by tech teams, the risk manager who asks awkward questions. I’m not at the fun brainstorming sessions, but I’m in your thoughts when things go wrong.

I get how I seem: a necessary hassle, not a cool teammate. You may remember all those audit controls I mention over and over again. They come from seeing what happens without them.

Asking for proof of encryption keeps data safe. Access reviews prevent small mistakes from becoming big problems. I just want to help turn “we probably should” into something solid. It can feel tough, but it’s there to support everyone.

I understand the looks

I see it when I share a new policy: that quiet sigh, "Do we really need this now?"

- Developers wonder if input validation can wait until after launch
- Management feels MFA (Multifactor Authentification) slows down the day
- Business thinks the certificate means we're covered

I live this too. Weeks spent on risk assessments and control checks, hoping it sticks, only to see shortcuts later because "we already passed the audit." I understand the frustration. But this work is about protecting what we all care about.

What I've learned from the hard moments

I don't bring this up to make life harder. I've seen the other side:

- A missed input check that lets in bad data in and harms customers
- A delayed patch leading to hours of recovery work
- A vendor trusted without safeguards, pulling everyone down

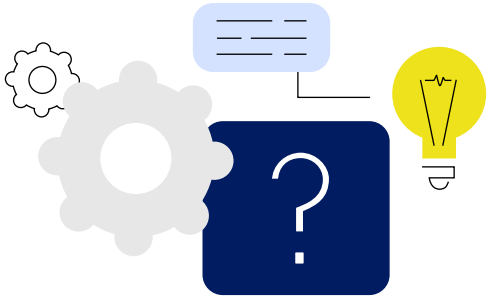
Compliance is like that extra check you do without thinking – locking the door twice, even on a quiet street. It's the developer refreshing API keys just in case. The sysadmin reviewing logs on a weekend. Quiet habits that keep things steady.

Collective responsibility ahead

This isn't just my job – it's everyone's. Every team member – not just audit attendees – needs to question assumptions, spot anomalies and choose risk awareness over shortcuts. Certified companies keep getting hit, but we don't have to be next. Think two steps ahead, ask for evidence behind every claim. Certificates win clients, but mindsets stop breaches.

Let's talk about it

Next time you spot the Ugly coming with questions, I hope you'll see a partner, not a roadblock. Ask me the 'why' behind it. Share your side. Together, we make things stronger. Because at the end of the day, this isn't about rules – it's about keeping what we've built safe for tomorrow.



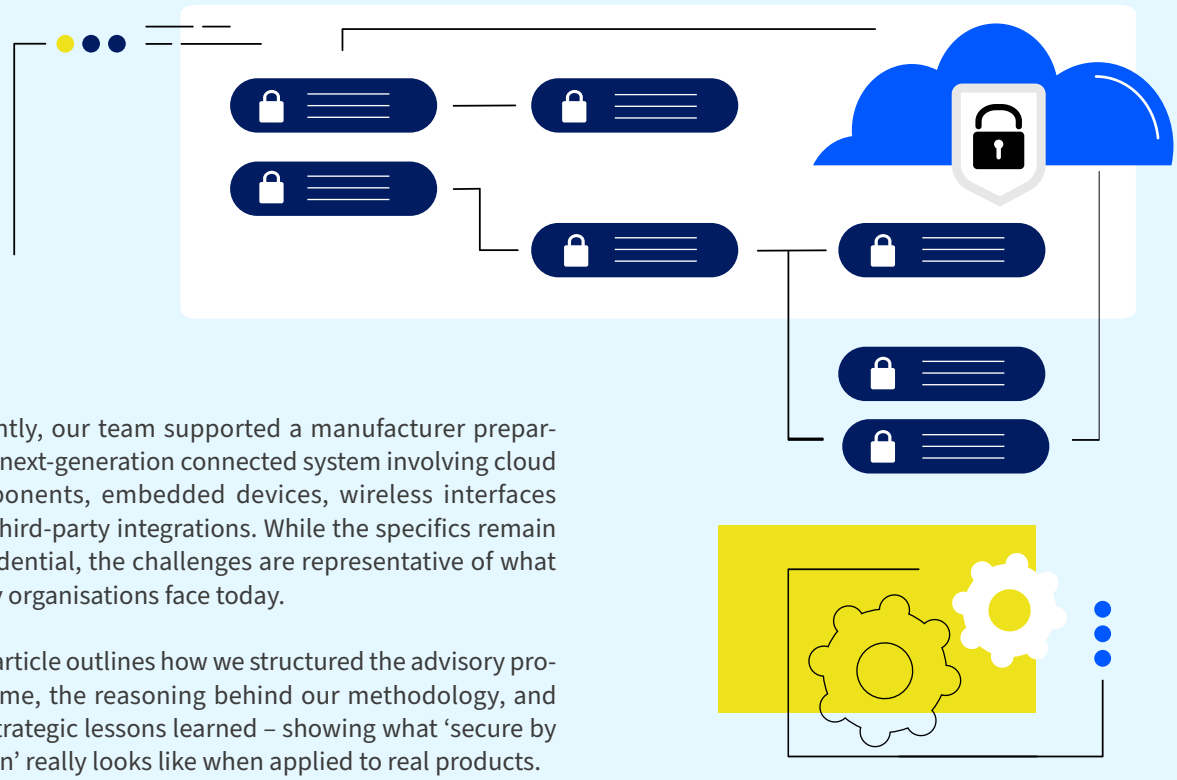
About José Francisco Agulló

José Francisco Agulló, our Quality Manager at ERNI Spain in Valencia, supports teams in building reliable, compliant and high-quality software solutions in regulated environments. With a background in industrial engineering and quality assurance, and as an ISO 27001 Lead Auditor, he combines hands-on engineering experience with a pragmatic approach to risk, security and quality.

A real-world cybersecurity use case under the EU Cyber Resilience Act

As digital products evolve into complex ecosystems – combining cloud services, embedded controllers, mobile applications and enterprise integrations – their cybersecurity requirements grow exponentially. With the EU Cyber Resilience Act and stricter MDR, ISO 14971 and IEC 81001-5-1 rules, manufacturers must prove their products are secure throughout their lifecycle.

By José Francisco Agulló, Quality Manager, ERNI Spain



Recently, our team supported a manufacturer preparing a next-generation connected system involving cloud components, embedded devices, wireless interfaces and third-party integrations. While the specifics remain confidential, the challenges are representative of what many organisations face today.

This article outlines how we structured the advisory programme, the reasoning behind our methodology, and the strategic lessons learned – showing what 'secure by design' really looks like when applied to real products.

Why lifecycle cybersecurity is no longer optional

Regulations like the Cyber Resilience Act fundamentally shift cybersecurity from a 'best effort' approach to a legal obligation. Manufacturers must now:

- Identify threats from design phase onward
- Provide secure defaults and update mechanisms
- Maintain an SBOM
- Assess exploitability pre-market
- Implement vulnerability management processes
- Document security controls in a technical file
- Support post-market monitoring throughout the entire lifecycle

For many companies, these are completely new disciplines. Our use case centred around creating a structured, defensible and traceable security programme that turns these obligations into clear engineering practices.

Building security from first principles: Threat modelling and risk analysis

The engagement began where every secure product should begin: with a deep understanding of the system and its risks.

Our philosophy: You cannot secure what you do not understand. And you cannot understand a system unless you model how data flows through it. We worked with engineering teams to create:

- Level 0 and Level 1 data flow diagrams
- Trust boundaries across cloud, embedded, mobile and enterprise interfaces
- STRIDE-based threat models for each component
- A risk matrix aligned with MDR + CRA Article 10

This stage forces clarity on:

- Where sensitive data lives
- Which modules introduce exploitable attack surfaces
- How components depend on each other
- Where security controls must be enforced
- What can go wrong, and how severely

Most importantly, it establishes the security intent, which auditors will later expect to see reflected in architecture and documentation.

Designing a secure architecture that supports the regulation

The second phase focused on defining a CRA-aligned secure architecture. Instead of adding security layers after development, the objective was to embed them into the foundation of the system. Our approach followed four core principles:

1. Zero Trust for connected products

Every interface – Bluetooth, USB, cloud API, workstation – must assume compromise by default.

2. Secure communication everywhere

We designed encryption strategies for:

- **Device ↔ cloud**
- **Device ↔ local workstation**
- **Device ↔ mobile**
- **Firmware update channels**
- **Enterprise system integrations**

3. Identity drives authorisation

Role-based access control and proper IAM design ensure that only the right entities can perform sensitive operations – critical for compliance under CRA Annex II.

4. Architecture must be explainable to auditors

This is key: a secure system that cannot be justified during an audit is not compliant. The result was a secure architecture specification, integrating access control, encryption, update policies, secure defaults and supply chain considerations.

Embedding security into the development lifecycle

One of the biggest gaps for manufacturers is that security is often not integrated into the SDLC. CRA and IEC 81001-5-1 demand:

- Secure coding requirements
- Reproducible development processes
- Vulnerability handling workflows
- Traceable evidence of testing and review
- SBOM generation and maintenance

Our role was to help the development team operationalise these concepts. We introduced:

- Security requirements per module
- SAST and DAST tools
- Manual secure code reviews
- Coding guidelines aligned with Annex II
- SBOM strategy and tooling
- Training for developers on threat patterns and mitigations

Philosophy: Security must become part of everyday engineering – not a ‘compliance activity’.

Validating the security posture through real testing

No matter how strong the design, a security programme is incomplete without hands-on validation. We performed:

- Penetration testing across cloud, device and workstation interfaces
- Fuzzing of USB and Bluetooth channels
- Robustness testing with malformed inputs
- Validation of logs, audit trails and timestamp integrity
- Review of update mechanisms and rollback paths

This phase serves two purposes:

- Demonstrate the effectiveness of the implemented security controls
- Generate the evidence required for CRA technical documentation

Testing uncovered several areas for improvement, but also confirmed that core architecture decisions were sound.

Producing the technical file for CRA and MDR

Regulators expect not just compliance, but demonstrable, traceable, auditable compliance. We helped the client assemble:

- Threat models
- Vulnerability assessments
- Security requirements
- SBOM documentation
- Update policy
- Risk assessment per Annex II
- Incident response plan
- Architecture dossier
- Validation and test reports

The final output was a complete CRA-aligned technical file, ready for notified bodies and pre-market evaluation.

Key insights from this case

This engagement highlights lessons valuable for any manufacturer developing connected products:

1. Cybersecurity must start before the first line of code
   Threat modelling saved months of rework and prevented architectural mistakes.

2. CRA is not just a regulation; it is a lifecycle mindset

Compliance is achieved through continuous processes, not isolated deliverables.

3. Documentation is as important as technical controls

Auditors must understand why decisions were made, not only what was implemented.

4. Secure development is a cultural shift

Teams must internalise principles like least privilege, defence-in-depth and secure defaults.

5. Testing needs to reflect real attacker behaviour

Fuzzing, malformed input testing and endpoint abuse scenarios reveal issues traditional testing misses.

6. A structured programme dramatically reduces regulatory risk

By aligning architecture, development, testing and documentation with the same framework, compliance becomes manageable – and predictable.

Conclusion: A practical path to secure, compliant and resilient products

The Cyber Resilience Act marks a turning point. Manufacturers of connected products must now demonstrate security throughout the entire lifecycle – design, development, validation and post-market. This use case illustrates how a structured advisory programme can guide organisations from uncertainty to readiness:

- **Clear threat understanding**
- **Secure architecture**
- **Secure coding practices**
- **Testing that reflects real-world threats**
- **Documentation that withstands regulatory scrutiny**

By treating cybersecurity as an engineering discipline rather than a checkbox exercise, companies can build products that are not only innovative but also trustworthy, resilient and compliant with the new European standards.

About José Francisco Agulló

José Francisco Agulló, our Quality Manager at ERNI Spain in Valencia, supports teams in building reliable, compliant and high-quality software solutions in regulated environments. With a background in industrial engineering and quality assurance, and as an ISO 27001 Lead Auditor, he combines hands-on engineering experience with a pragmatic approach to risk, security and quality.

The Bad: A pentester's day – Breaking things so others can fix them

Pentesting rarely begins with alarms. It starts with a quiet request to test assumptions. This article takes you inside an offensive security engagement, where discipline meets intuition, logic flaws matter more than exploits, and real security is built by proving what can break. It shows why this matters more than policies alone.

By Iván Martínez, Software Developer and Pentester, ERNI Spain

The request never arrives with drama. No alarms. No red lights. Just an email. Usually short. Polite. A scope. A date. A reminder about the rules of engagement. Sometimes a line that reads: "We want to know how bad it really is."

That's when I smile. Not because I enjoy chaos for its own sake, but because this is the moment where theory meets reality. Someone, somewhere, has decided to stop trusting assumptions and start testing them. And that decision – more than any firewall or policy – is the first real act of security.

The call to break

When I receive a pentesting request, the first thing I feel is responsibility. This is not a game, even if the tools sometimes look like toys to the uninitiated. These are real systems. Real data. Real reputations. My job is not to show how clever I am, but to show how fragile certainty can be.

I read the scope carefully. What's in bounds matters as much as what's out. IP ranges. Applications. Environments. Credentials – or the lack thereof. Black box, grey box, white box. Every choice shapes the story that will follow.

Methodology comes first. Always. Before touching a single system, I align myself with a framework: the OWASP Testing Guide, PTES, sometimes MITRE ATT&CK to map tactics and techniques. This isn't improvisation. It's discipline. It's how I make sure that whatever I find can be explained, reproduced, and – most importantly – fixed.

Reconnaissance: Learning the town

The first phase is quiet. No exploits. No payloads. Just observation. Enumeration. Listening.

I map the attack surface like a stranger walking through a new town, noting which doors are locked, which windows are open, and which lights are still on at night. DNS records reveal forgotten subdomains. Open ports whisper about services that should have been retired years ago. Error messages say far more than they should.

This phase is almost meditative. Calm. It's about understanding how the system presents itself to the world – and how much it reveals without being asked. I document everything. Screenshots. Versions. Timestamps. Because later, every detail will matter.

The first push

Exploitation never starts with brute force. It starts with a feeling. A pause that's just a bit too long. A response that doesn't quite line up with the logic I expect. An endpoint that behaves correctly in isolation, but oddly when chained with another. This is the part that's hardest to explain to anyone outside the craft: sometimes you don't see the vulnerability yet – you sense it.

I slow down here. This is where pattern recognition takes over. Years of broken systems whispering in the back of my mind. I replay the flow of the application in my head, not as code, but as intent. What the developers meant it to do. Where trust changes hands. Where assumptions are made.

A parameter behaves too generously. A state transition skips a step. Authentication does its job – but authorisation follows a different logic entirely. These aren't textbook vulnerabilities. They're personality traits of this specific system. And that's what makes them dangerous.

I test gently at first, nudging the logic rather than smashing it. One request becomes two. Two become a sequence. Each response tells me whether I'm getting warmer or colder. When the system reacts in a way it shouldn't, I feel it immediately. Not excitement – clarity.

This is the dangerous moment. The line between controlled testing and real damage is thin. Every payload is crafted. Every step reversible. I don't exploit to dominate – I exploit to understand.

Sometimes the system invites me deeper. A role that was never meant to be chained becomes a stepping stone. A token meant for one context suddenly works in another. Privileges stack not because of a bug, but because of a story no one finished writing. This is where logic flaws reveal themselves: unique, non-repeatable anywhere else, born from business rules rather than bad code. Inside the system, I feel the full weight of misplaced trust.

Inside the walls

There's a moment, once you're inside, where everything goes quiet. This is where discipline matters most.

There's adrenaline, yes – but it's tempered by restraint. I don't empty databases. I don't alter records. I take the minimum proof required to demonstrate impact and no more. A screenshot. A query result. A controlled action that shows reach without harm.

I think about the developers who built this system. The pressure they were under. The trade-offs they had to make. Most logic flaws aren't born from incompetence – they're born from complexity.

And I think about what this access means beyond the technical layer. Regulatory exposure. Business continuity. Trust. This isn't just a vulnerability – it's a risk narrative, and my job is to tell it accurately.

Documentation: Turning insight into action

The pentesting doesn't end when access is achieved. It ends when understanding is transferred.

I document every finding as a story with a beginning, a middle and a consequence. Not just what is broken, but why it breaks in this system, and under which assumptions it becomes exploitable. I describe the attack path in human terms, mapping technical steps to business logic.

Recommendations matter here. Generic advice helps no one. Each fix is contextual: tightening a trust boundary, enforcing state validation,

separating roles that should never overlap, adding server-side checks where client-side assumptions once lived. I explain not just how to patch the issue, but how to prevent its class from reappearing.

And I always think ahead to the retest. A fix isn't real until it's verified. I outline what success should look like: which paths must now fail, which responses should change, which assumptions should no longer hold. A good retest doesn't just confirm closure – it restores confidence.

Severity is assigned with care, grounded in impact and likelihood, not theatrics. A critical issue in a lab is not the same as a medium one in production. Context is everything. The report is not a trophy. It's a map.

After the dust settles

When the engagement ends, there's no victory lap. I sit in the debrief. I answer questions. I explain paths taken and paths avoided. Sometimes I see relief. Sometimes disbelief. Often gratitude.

And then I move on. Another system. Another scope. Another quiet email. I don't stay to watch the fixes. That's not my role. But I know that somewhere, defences will be improved because I showed how they could fail.

Why I do this

I live in the uncomfortable space between trust and proof. I feel the tension of being invited to attack – and the responsibility that comes with it. I enjoy the challenge, yes, but I respect the stakes far more.

Without offensive security, protection is theoretical. Assumed. Untested. With it, security becomes real.

About Iván Martínez

Iván Martínez, Software Developer and Pentester at ERNI Spain, is working on MedTech projects in the biotechnology and healthcare sector. He specialises in developing and optimising drivers for Point of Care diagnostic devices, enabling seamless integration with Laboratory Information Systems using standards such as HL7 and ASTM. Driven by curiosity, he combines software engineering with a strong interest in cybersecurity and secure systems design.

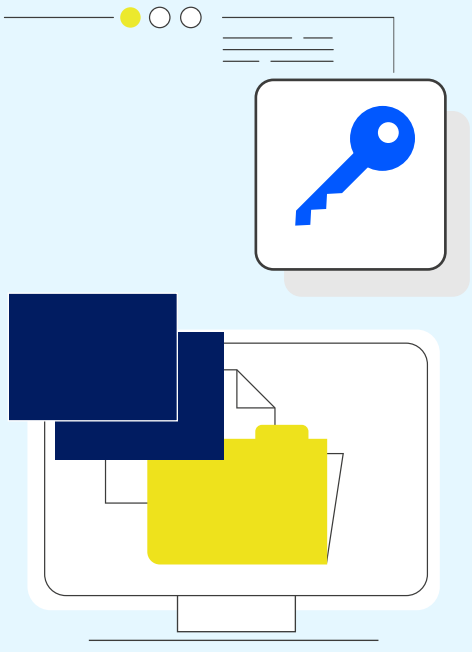
Bringing security to life: A real-world penetration testing journey

Modern connected systems increasingly blend cloud services, mobile applications, wireless protocols and embedded hardware into a single operational ecosystem. This convergence expands functionality – but also dramatically enlarges the attack surface.

By Alessandro Palermo, Senior Consultant, Programme Manager and Product Owner, ERNI Switzerland

In a recent cybersecurity engagement, our team was asked to assess the security posture of a multi-layered digital access system used in industrial and commercial environments. Although the specifics of the customer remain confidential, the system combined:

- A cloud platform managing user roles, permissions and logs
- A mobile application used by technicians and operators
- An IoT gateway enabling device connectivity
- A hardware control unit communicating via CAN bus
- Bluetooth Low Energy (BLE) communication for local interaction



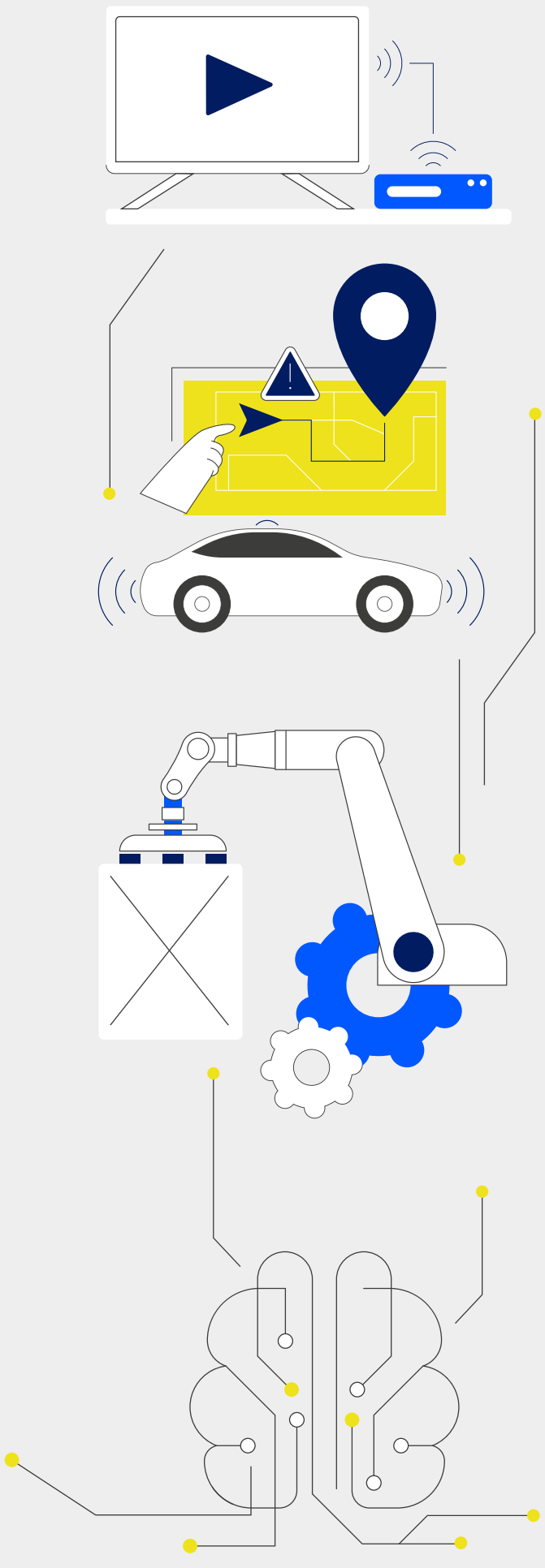
The challenge was clear: How do you evaluate the security of a system where cloud logic directly influences physical motion? This article outlines the methodology, the reasoning behind our testing philosophy, and the lessons learned – demonstrating how real-world penetration testing must evolve alongside the systems it aims to protect.

Why having a holistic methodology matters

Many organisations still treat penetration testing as a set of isolated exercises: one for the web app, another for mobile, another for embedded hardware. However, modern threats do not respect departmental boundaries. Attackers pivot fluidly:

- from mobile to cloud,
- from cloud to physical devices,
- from wireless channels to embedded controllers.

For this reason, our approach is built around a central principle: Security must be evaluated as a system, not as components. This philosophy drove the design of our test plan. Rather than running five separate assessments, we created a single overarching methodology that mapped interdependencies, trust boundaries and potential pivot paths across the entire ecosystem.



Phase 1:

UNDERSTANDING THE SYSTEM THROUGH THREAT MODELLING

Before touching any tool, we begin with context. Therefore, we conducted structured threat modelling workshops. In the following overview, we show what we aim to identify in such workshops:

- Primary assets (physical access, user identities, operational logs, configuration data)
- Threat actors (external attackers, rogue employees, malicious technicians, compromised mobile devices)
- Attack motivations (unauthorised door opening, privilege escalation, sensitive data exposure, disruption of operations)
- Critical trust boundaries (cloud-to-device, mobile-to-API, BLE-to-controller, etc.)

This phase guides the rest of the engagement by answering a key question: If we were attackers with realistic constraints, where would we strike first? The outcome was a prioritised list of technical surfaces that attackers could exploit – allowing us to allocate testing effort where it mattered most.

Phase 2:

TESTING THE CLOUD APPLICATION AND APIS

The cloud layer is the 'brain' of the system. It manages authentication, authorisation and the orchestration of physical commands sent to field devices. Our testing combined the OWASP Cloud-Native Application Security Top 10 with scenario-driven threat modelling:

- Can a non-privileged user escalate to administrative roles?
- Could misconfigured APIs allow unauthorised control actions?
- Do cloud misconfigurations expose sensitive operational data?
- Could flaws allow remote manipulation of physical devices?

We applied:

- Injection testing (SQL, command, LDAP, XXE)
- Broken Access Control analysis

- Hardening validation (headers, CSP, CORS, TLS configuration)
- Secrets exposure checks (e.g., API keys, JWT flaws)
- Logging and monitoring evaluation
- Identity & Access Management (IAM) assessment
- File upload abuse scenarios

The philosophy here is simple: Cloud vulnerabilities often magnify physical impact. A compromised dashboard can be more dangerous than physical tampering.

Phase 3:
ASSESSING THE MOBILE APPLICATION

The mobile app served as the primary operational interface for technicians.
This made it both a productivity tool and a potential attack vector.

Using the OWASP Mobile Security Testing Guide (MSTG), we evaluated:

- Secure local data storage
- Credential handling and secrets management
- JWT validation and token integrity
- Runtime security controls (root/jailbreak detection, anti-tampering)
- API interaction security
- Reversing resistance (code obfuscation, binary hardening)
- Dependency analysis and cryptography review

Our guiding philosophy: A mobile app should assume that the device it runs on may already be compromised. This mindset uncovers issues such as hardcoded credentials, insufficient validation strategies or insecure session handling.

Phase 4:
EVALUATING THE HARDWARE AND CAN BUS COMMUNICATION

The embedded control system and CAN bus formed the physical backbone of the solution. Unlike traditional IT services, these components interact with motors, sensors and actuators – meaning security flaws can lead to real physical outcomes. Our testing included:

- Bus traffic capture and protocol analysis
- Message injection and manipulation
- Fuzz testing to evaluate resilience
- Tampering attempts against housings and connectors
- DoS testing to understand fault tolerance
- Review of ECU behaviour under stress
- Assessment of anti-tampering measures

Our philosophy in this domain: Physical systems rarely fail dramatically – they fail quietly. Understanding their behaviour under unexpected inputs is essential. This allowed us to validate how the device responded to flooding, malformed frames and unexpected state transitions.

Phase 5:
EXAMINING THE IOT GATEWAY AND CONNECTION PATHS

As a bridge between local hardware and remote cloud services, the IoT gateway required a different perspective. We approached it with OSSTMM principles, focusing on:

- Enumeration of network surfaces
- Port and service exposure
- Device hardening
- LAN and GSM communication analysis
- Protocol behaviour under adversarial conditions

A key element of our philosophy here: A secure gateway is invisible.
Proper hardening makes the attack surface disappear.

Phase 6:
TESTING BLUETOOTH LOW ENERGY (BLE) COMMUNICATIONS

BLE enables convenient local interaction – but also introduces radio-based attack opportunities. Our custom BLE methodology included:

- Scanning and device enumeration
- Pairing and authentication testing
- Encryption validation
- Replay and injection attacks
- Signal resilience testing under interference
- Fuzzing of GATT services and characteristics

Our principle in this layer: Wireless features must be treated as public entry points – regardless of their intended range.

Key insights from the assessment

Although individual findings vary between systems, several insights consistently emerge in cyber-physical ecosystems:

1. Access control is often the weakest point
   Role misconfigurations, insecure API design and insufficient server-side checks create opportunities for privilege escalation.
2. Secrets tend to leak where developers least expect them
   Hardcoded API keys, insufficient JWT validation and unprotected configuration files are common across industries.
3. Mobile devices cannot be trusted implicitly
   Rooted device detection, secure local storage and code obfuscation remain essential.
4. Physical devices require resilience, not perfection
   Fail-safe behaviour, tamper detection and robust rate limiting on communication buses are more important than complex cryptographic designs.
5. Monitoring and incident response are often overlooked
   A system may detect unauthorised access – but without alerting, nothing happens.

Building a philosophy of practical security

Our methodology is not tool-driven; it is impact-driven. At each stage, we ask:

- What is the real-world consequence of this vulnerability?
- Can this flaw be exploited with realistic attacker capabilities?
- How does this component influence the behaviour of the broader system?
- Does the system fail safely or dangerously?

This perspective enables us to adapt testing depth dynamically across components, ensuring the highest value for the customer.

Conclusion: Security across the digital-physical boundary

As industries increasingly rely on connected devices, the boundary between cybersecurity and physical safety continues to blur. A cloud misconfiguration can open a physical door; a mobile app flaw can escalate privileges; a CAN message can disrupt operations. Our recent engagement demonstrates that effective security assessments must integrate cloud, mobile, IoT, wireless and embedded testing into a unified methodology. Organisations that adopt this holistic perspective gain more than a vulnerability report – they gain clarity, resilience and confidence in the systems that keep their operations running.

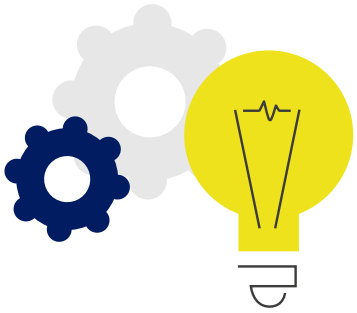
About Alessandro Palermo

Alessandro Palermo, Senior Consultant, Programme Manager and Product Owner at ERNI Switzerland, leads complex, large-scale digital products for global clients. PMP-certified, he specialises in agile leadership, product vision and stakeholder management, guiding interdisciplinary teams from concept to global rollout. His work bridges strategy and execution, with a strong focus on customer value and sustainable delivery.

When the dust settles: What remains standing in the new digital frontier

In classic Westerns, the story doesn't end with the last shot. It ends after the dust settles – when the town decides its future. Modern cybersecurity is the same: it's not about the breach, the audit or deployment – it's about what happens next and how teams shape resilient, secure systems.

By David Soto Dalmau, Cybersecurity Principal, ERNI Spain



Throughout this issue, we have crossed complex territory: secure development, regulation, offensive testing, legal frameworks, connected systems, physical devices, cloud platforms, code, processes and people. We have explored real threats, growing obligations, and architectures that can no longer afford naïveté. Yet if one idea runs through every page, it is this: Security is not a state – it is a way of operating.

The myth of the 'secure' system

For years, the industry has quietly chased a promise: that if we do enough, we will eventually reach a point where a system is 'secure'.

That point does not exist. Not because we lack tools, standards or talent, but because living systems – those that evolve, integrate, connect and scale – are never static. Context changes. Actors change. Incentives shift. And above all, trust is exploited in new ways.

Maturity does not arrive when vulnerabilities disappear. It arrives when they no longer surprise us.

Three perspectives, one shared responsibility

Across this magazine, we have given shape and voice to three forces that have always been present:

- Those who build with intent and discipline
- Those who set boundaries, so lessons are not forgotten
- Those who break assumptions to prove that reality does not negotiate

The common mistake is to treat these as separate functions. The real risk appears when they become silos. When development advances

without validation, confidence becomes fragile. When regulation exists without technical understanding, it turns into noise. And when attack is practised without context, it becomes spectacle.

Effective security emerges only when these perspectives challenge one another, correct one another and force each other to improve. It is not comfortable. But it is deeply professional.

From compliance to conviction

Regulations like the Cyber Resilience Act are not the destination – they are a signal. They reflect what many organisations have already learned the hard way: security cannot depend on isolated heroics or last-minute efforts before an audit.

Compliance is necessary. But it is not enough. Resilient organisations are not defined by passing assessments, but by their ability to explain – clearly and honestly – why their systems are designed the way they are, which risks they consciously accept, and how they will respond when something fails.

Because something will fail. The question has never been if, but how prepared we are when it does.

The real shift is not technical

After all the tools, architectures, threat models and tests, one uncomfortable conclusion remains:

The most meaningful advances in cybersecurity are not technological – they are cultural. They happen when:

- Developers understand that security is part of quality
- Compliance professionals understand the systems they govern
- Offensive teams are seen as allies in learning, not bearers of bad news

They happen when the conversation shifts from “Who failed?” to “What did the system teach us this time?”

When the dust settles

In Westerns, the truly civilising moment is not the duel – it is when someone realises they no longer need to keep a hand hovering over their gun. In the digital world, that moment arrives when security stops being a constant reaction and becomes an integrated, everyday practice – almost invisible.

Not because risk has disappeared, but because the organisation has learned to live with it without denial.

That is the territory towards which everything you have read here points. Not a world without threats, but one where we understand the game, recognise its real rules and take responsibility for the role we play. The dust settles. The system remains alive. And the real work – the meaningful work – continues.

About David Soto Dalmau

David Soto Dalmau is Cybersecurity Practice Area Lead at ERNI in Barcelona. As a lifelong ethical hacking and CTF enthusiast, he combines expertise in cryptography, penetration testing and risk management with teaching and fostering a culture of continuous learning.

About ERNI

ERNI stands for Swiss Software Engineering. What are we really interested in? How we can support you and your employees better than any other company in developing and marketing software-based products and services. Our global platform for software development, in combination with a sound understanding of the market, forms the framework for our customers' success. Our team also implements complex projects, empowers people and delivers outstanding customer solutions in the shortest time. We apply the Swiss mentality with behaviours such as consensus building, pragmatism, integration, reliability and transparency on a global scale – and have done so since our foundation in 1994 together with our great team, which forms the basis for your successful software projects. Today, the ERNI Group employs more than 800 people worldwide.

About .experience

In this magazine, which is published a couple of times per year by ERNI, we provide information about important learning experiences that we have had in our daily work in the areas of collaboration, processes and technology.

Imprint

Issue 1/2026

ERNI
Swiss Software Engineering
betterask.erni

Publisher
ERNI Management Services AG

ERNI Locations

ERNI Schweiz AG
Bern
Zurich
Lucerne
Lausanne
Basel

ERNI Consulting España S.L.U.
Barcelona
Madrid
Sant C. del Vallès

ERNI Development Center Spain, S.L.
Valencia

ERNI (Germany) GmbH
Frankfurt
Munich
Berlin
Stuttgart

ERNI Development Center Philippines Inc.
Manila

ERNI Development Center Romania S.R.L.
Cluj-Napoca

ERNI Singapore PTE. LTD.
Singapore

ERNI (Slovakia) s.r.o.
Bratislava

ERNI USA
New York

Contact

ERNI Management Services AG
Löwenstrasse 11 I 8001 Zurich
Email: marketing@betterask.erni
Phone: +41 58 268 12 00
Web: www.betterask.erni

ERNI on the social networks