THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference

Driving Strategic Value Through Open Source

OSPOs and R&D organizations in the era of the CRA

Who Are We?

Georg Kunz

Open Source Program Office @ Ericsson

David Östman

General Manager @ Ericsson Software Technology Sweden

EU Cyber Resilience Act (CRA)

- Fundamentally about…
  - All products with software (digital elements)
  - CE marking for software
  - Product-based risk assessment
  - Due diligence for software components
  - Not shipping software with known exploitable vulnerabilities
  - Reporting of identified vulnerabilities
  - Sharing vulnerability fixes with upstream

The Cyber Resilience Act

Selected obligations of manufacturers...

Obligations for Manufacturers

- Article 13 (5) – Exercise due diligence

  For the purpose of complying with paragraph 1, manufacturers shall exercise due diligence when integrating components sourced from third parties so that those components do not compromise the cybersecurity of the product with digital elements, including when integrating components of free and open-source software that have not been made available on the market in the course of a commercial activity.

Obligations for Manufacturers

- Article 13 (6) – Sharing vulnerability fixes

  Manufacturers shall, upon identifying a vulnerability in a component, including in an open source-component, which is integrated in the product with digital elements report the vulnerability to the person or entity manufacturing or maintaining the component, and address and remediate the vulnerability in accordance with the vulnerability handling requirements set out in Part II of Annex 1. Where manufacturers have developed a software or hardware modification to address the vulnerability in that component, they shall share the relevant code or documentation with the person or entity manufacturing or maintaining the component, where appropriate in a machine-readable format.

Obligations for Manufacturers

- Article 13 (8) – Effective vulnerability handling

  Manufacturers shall ensure, when placing a product with digital elements on the market, and for the support period, that vulnerabilities of that product, including its components, are handled effectively and in accordance with the essential cybersecurity requirements set out in Part II of Annex I.

Obligations for Manufacturers

- Exercise due diligence
- Sharing vulnerability fixes
- Effective vulnerability handling

What do they have in common?

All benefit from strong upstream collaboration

Obligations for Manufacturers

- Exercise due diligence
  - Participation provides better insight
  - Supply chain resilience through maintainer relationships

CRA compliance is less risky and costly with strong upstream engagement

  - Efficient collaboration for both project and user

- Effective vulnerability handling
  - Keep components and their dependencies to date
  - Early involvement in vulnerability response, e.g., security team participation

One Year Later: From Vision to Operating Model

Last year I told you we were starting

- The Power of the Internal Trio: Developer, R&D Management, OSPO
  - *Developer: key to success — technical insight and community presence*
  - *R&D Management: strategic alignment and mandate*
  - *OSPO: policy, compliance, governance*
  - *Three perspectives, needed together to move fast in a large company*
- Our journey to Valkey, proof of what the trio enables
- Perseverance, two years of steady effort to drive a strategic shift in the company
- A commitment/shift to engage more directly in Linux and join LF Yocto project rather than relying on third parities

This year — what it became

- Zooming into the Linux effort
- An in-house Linux distribution, built on Yocto
- The team and the journey
- The business value
- CRA-readiness by design

The Journey: Commitment to Operating Capability

We thought we were building a team

- A small dedicated team — the right skills, the right mindset, the Open-Source way
- A commitment to engage directly in Linux and join Yocto, rather than relying on third-party vendors
- Building momentum

We were actually building a capability

- Demand grew fast — extensive hiring — a good problem to have
- Team tripled in size
- Running delivery and support for the company's two most critical products
- Mandate, authority, upstream by default

- The team isn't just bigger — it has a mandate – operates upstream since start

Going In-House: Reclaim Control of the Stack

No man-in-the-middle, we sit with maintainers, silicon vendors, and the upstream community

What We Gave Up

- A vendor support contract
- Someone else's throat to choke
- The comfort of "not our problem"
- The illusion of risk transfer
    - Shifted risk — on paper

What We Gained

- Internal competence that compounds
- Upstream leverage
- Timing of releases on our schedule
- Control over what's in every image
- Direct vulnerability response
- Cost efficiency was a positive outcome — not the driver

This was about technology leadership, autonomy, and platform alignment

Third-Party Vendor Model

We didn't just change who did the work — we changed whose interests the work served

- License scales with deployment
- Annual support subscription scales with time
- Consultancy scales with every integration, extra BSP, maintenance activity, and long-term support
- Dependent on the vendor's direction, model rewards complexity
- Upstream engagement can present challenges for the business model
- Fixed release cadence, on their schedule, not ours
  - Customer demands are increasing, requiring more frequent releases for vulnerability management
  - Agentic AI tools are starting to discover CVEs at scale — more CVEs, more releases, more pressure
- The CRA will require effective, timely vulnerability handling and security updates by law, we're the manufacturer

The Cost Story: Direct, Indirect, Compounding

We expected in-house to cost more. It doesn't — we see predictable internal spend

~30-40% cost reduction in direct costs

- Uplift cadence collapse: Years → Months → Weeks
- More frequent releases with smaller deltas, no vendor contract re-negotiation
- Uplift lead times decrease with earlier pre-testing
- Tailored distributions for our products — we ship only what runs, optimize for legacy products
    - Smaller attack surface → fewer CVEs to triage
    - Tailored distributions → fewer field issues (TRs)
- We carry less, and handle what we do carry better
- Meet Customer and CRA obligations

Compounding: Every week, more value

- Third party adds natural delay in communication
    - Early detection of changes affecting our applications, and time to act on them, direct line to silicon vendors
- Linux/Yocto Center of Excellence, In-house authority to act, no vendor ticket queue
    - Centralized FOSS handling cross portfolio → reduced product handling cost and burden
    - Bridge Yocto's speed of change with our products' pace
- The cost we used to pay a vendor is now paying engineers who also generate upstream leverage and product-specific value
- Other groups in company are lining up to take this distribution

Upstream First: Beyond the Supplier Model

Team built upstream-first as a founding principle — direct access changes everything

- Deeper engagement in Linux/Yocto built our understanding of the long-term value
- Without a third party, we have direct contact with the community — patches, conversations, decisions
  - We now sit directly with Linux maintainers and silicon vendors, opening our eyes to new possibilities
- We learned the community's pain points — what to build, how and where to contribute
- CRA and vulnerability management — top of mind for us and our customers, and easier to handle when we're upstream
- Every dependency we remove is a CVE we don't have to triage

Kernel CVE triage tools by Daniel Turull, Ericsson Linux team

- Yocto: Tailor and build your own embedded distribution
- Kernel CNA → CVE explosion, mostly false alarms for a build
- Script uses SPDX data from the Yocto build to compare CVE metadata (kernel.org) with what was actually compiled
  - 70–80% fewer CVEs to triage for a typical build
  - Upstreamed to Open Embedded and Yocto (Link)
- CVE fixes: update the component or backport the fix, we are working on improving/automating both
- This is what our Yocto membership looks like in practice, our goal from the start, not just a logo on the Yocto webpage
- We built it to save our team time, upstream multiplies the value. Others save time too, and the CVEs they fix benefit us back, faster CRA-readiness for everyone

What the CRA Requires, What We Already Do

We built it because it was the right thing to do — and the CRA now confirms it

Due Diligence

- Tailored distributions, fewer components, less due diligence
- Centralized FOSS handling, one consistent process
- Upstream presence, we know the maintainers, can impact
- Due diligence isn't a scan — it's a relationship

Sharing Fixes

- Upstream first, avoid private forks
- Machine-readable outputs (SBOM, SPDX, VEX)
- We commit to fixing and improving Yocto long-term, not just for our needs
- Daniel Turull's tool contributions are Article 13(6) in action
- We share fixes because we built the team that way — not because the CRA asked

Effective Handling

- Customer demands made vulnerability management non-negotiable, long before the CRA
- Uplift cadence compatible with the full support period, and in our control
- 70–80% triage reduction makes it sustainable
- Daily triage, in-house authority to act

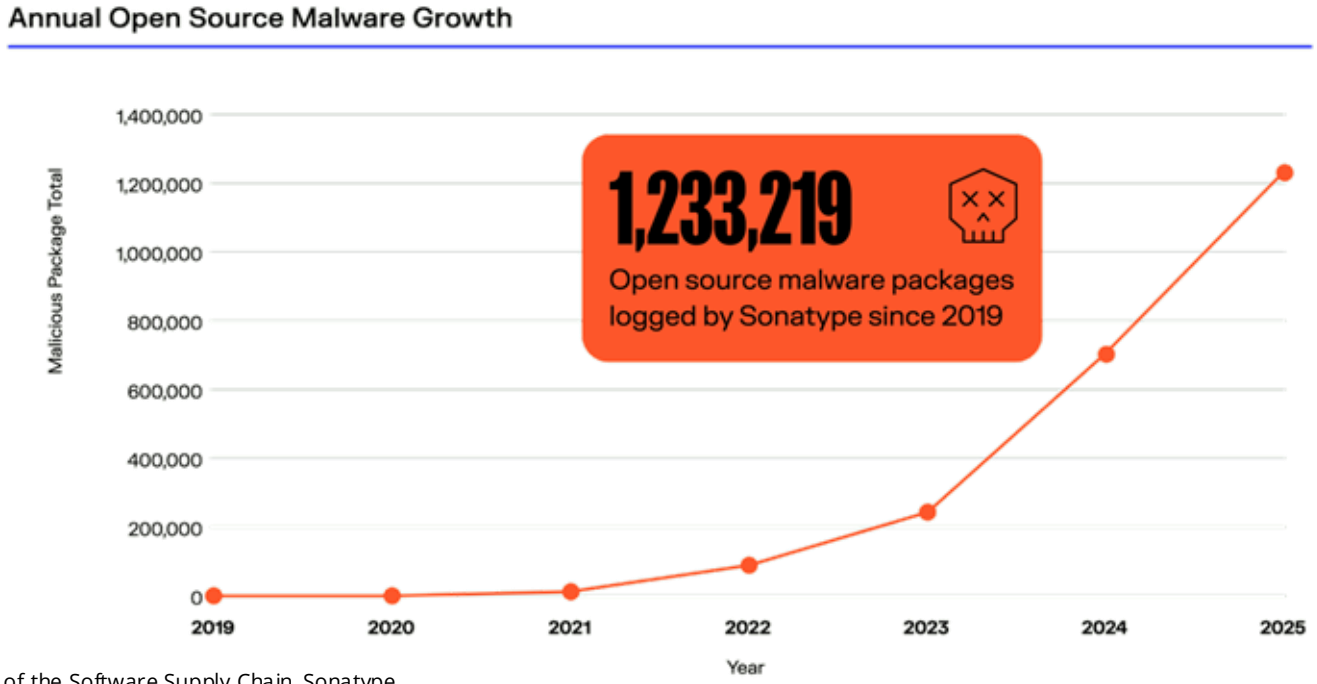
- Effective vulnerability handling isn’t only about policy, it's daily meetings, scripts and tools, and a team with the authority to act

Founding Principles, Now Required

We made this decision for technology leadership reasons — we didn't know the CRA was coming in its current form

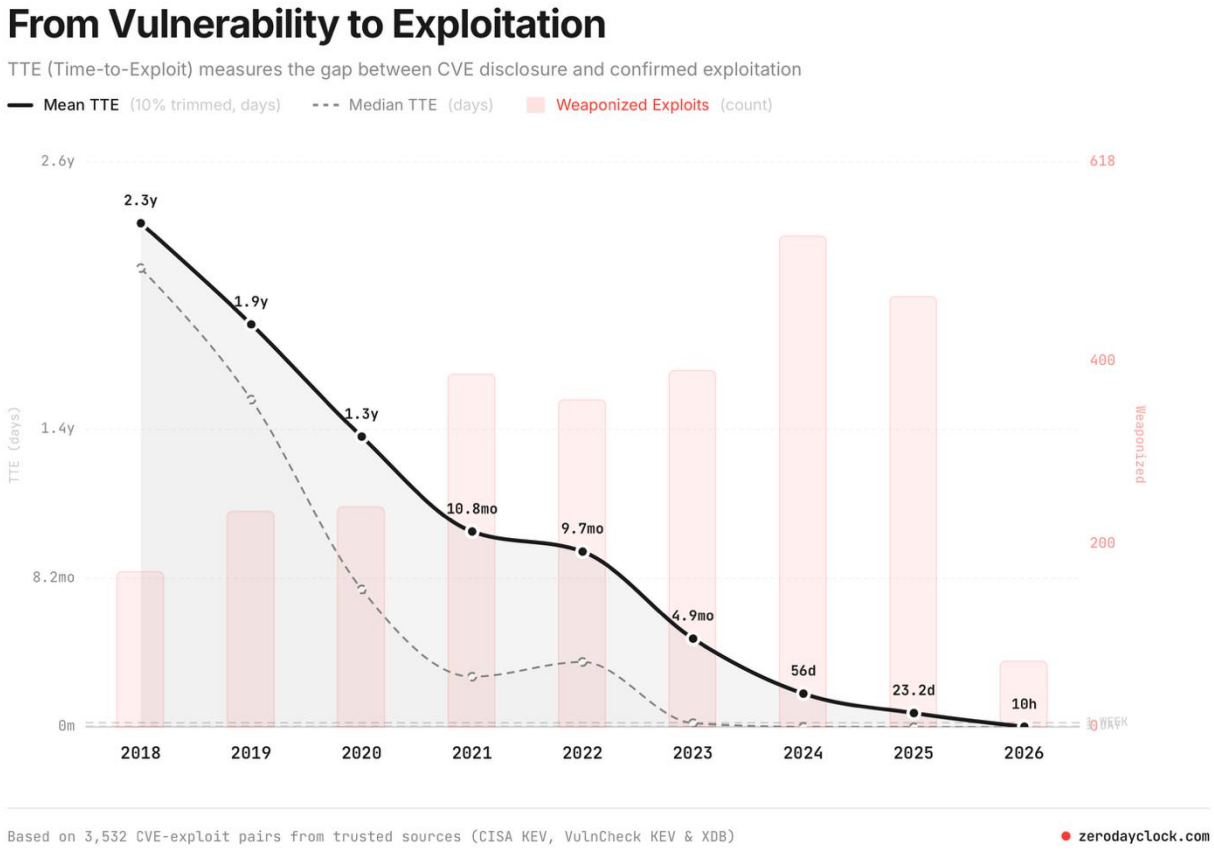
- Every argument we made internally — control, speed, competence, upstream engagement — turns out to be what the regulation now requires
- We see direct cost savings — and indirect savings that compound every week
- Upstream gives us strategic leverage — direction, decisions, and our own technology destiny
- Vulnerability management is key in CRA and non-negotiable for our customers — our model meets that
- In-house upstream-first isn't just cost-effective — it’s more compliant by design

Emerging Challenge: Vulnerability Overload



Source: 2026 State of the Software Supply Chain, Sonatype

Emerging Challenge: Vulnerability Overload



Emerging Challenge: Vulnerability Overload

- AI-assisted identification of vulnerabilities

- Challenges
  - Huge number of vulnerabilities, exceeding current handling practices
  - Duplication of costs and effort of users
    - Many users scanning the same open source projects, burning tokens
  - Duplication of costs and effort of maintainers
    - Maintainers receive duplicate vulnerability reports
  - Potential for **significant** friction in the ecosystem

- How do we tackle this as a community, industry, and ecosystem?

Call to Action

A network of partners with the shared goal to…

- Actively engage with upstream projects to mitigate security issues
- Create synergies and efficiency gains through mutual support and splitting of work
- Discuss mitigation of fundamental security concerns in critical projects
- Keep open source secure, interoperable, and non-fragmented

… in a safe environment.

How about joining us?

Conclusions

- CRA compliance is less risky and costly with strong upstream engagement
- Teams built with open-source DNA scale better and meet the regulation naturally
- Direct access changes everything — patches, conversations, decisions
- Strategic engagement lets us shape our own future — direction, decisions, destiny
- We see direct cost savings and indirect savings that compound every week
- Collaboration is key to secure open source

Q&A