LINUTRONIX

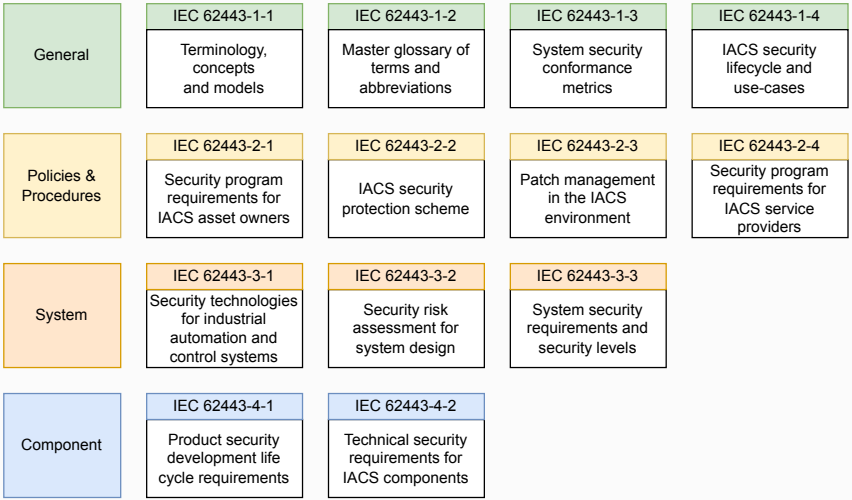
10. March 2026

Leveraging Open Source for IEC 62443-Compliant Embedded Systems

Dr. Florian Kauer, florian.kauer@linutronix.de

Introduction into IEC 62443 and the Cyber Resilience Act

LINUTRONIX



LINUTRONIX

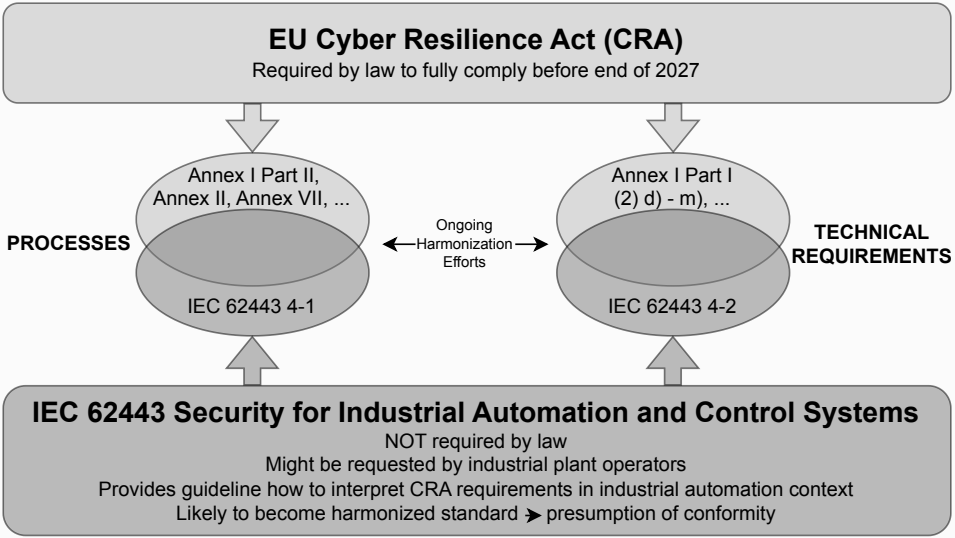
IEC 62443-4-1 - Secure product development lifecycle requirements

LINUTRONIX

IEC 62443-4-2 - Technical security requirements for IACS components

Introduction into IEC 62443 and the Cyber Resilience Act

LINUTRONIX



LINUTRONIX

Open Source in Industrial Automation

Collaborate on the Core, Compete on the Edges

LINUTRONIX

Open Source in Industrial Automation

Collaborate on the Core, Compete on the Edges

But is that feasible if compliance and certification is required?

LINUTRONIX

Open Source in Industrial Automation

Collaborate on the Core, Compete on the Edges

But is that feasible if compliance and certification is required?

Idea: Certify a demonstrator built from open-source components

LINUTRONIX

Case Study: IEC 62443-4-2 Certification of IGLOS Secure Beacon

- Embedded device with simple application:
  Remote control of blinking LED
- Locked-down and hardened Linux system
- Fulfills all 57 applicable security requirements from 4-2
- Certified by TÜV SÜD according to IEC 62443-4-2 in Q1 2025
- Serves as blueprint to implement more complex applications

Industrial IT Security
TÜV SÜD
IACS Component
Secure Product Development Lifecycle assessed & monitored according to IEC 62443-4-1
Technical Security Requirements assessed according to IEC 62443-4-2

How is Open Source Relevant for a Product Certification?

Traceable Documentation

Technical Requirements

External Components Management

How is Open Source Relevant for a Product Certification?

Traceable Documentation

Technical Requirements

External Components Management

LINUTRONIX

Traceable Documentation is the Primary Evidence for Certification

IEC 62443-4-1 SM-1

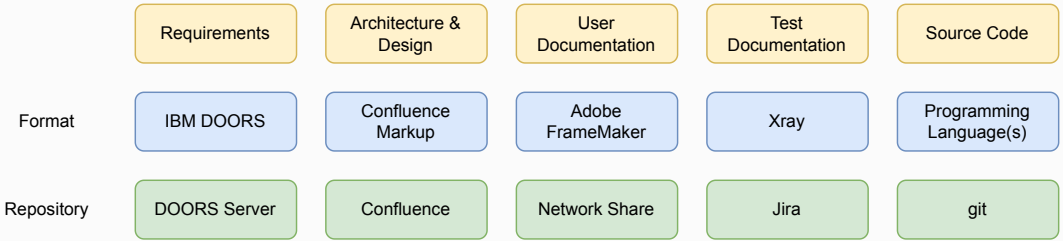
A [...] process shall be documented and enforced [...] that include [...] b) product description and requirements definition with requirements traceability [...]

IEC 62443-4-1 SI-1

A process shall be employed [...] including [...] d) review of the implementation and its traceability to the security capabilities defined to support the security design [...]

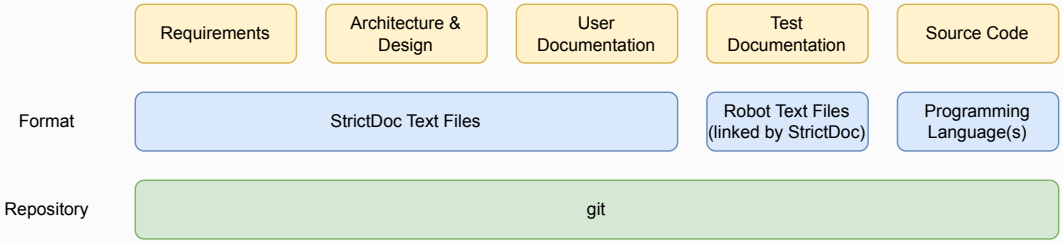
LINUTRONIX

Examplary Toolset for Product Development Documentation



LINUTRONIX

Product Development Documentation with StrictDoc



LINUTRONIX

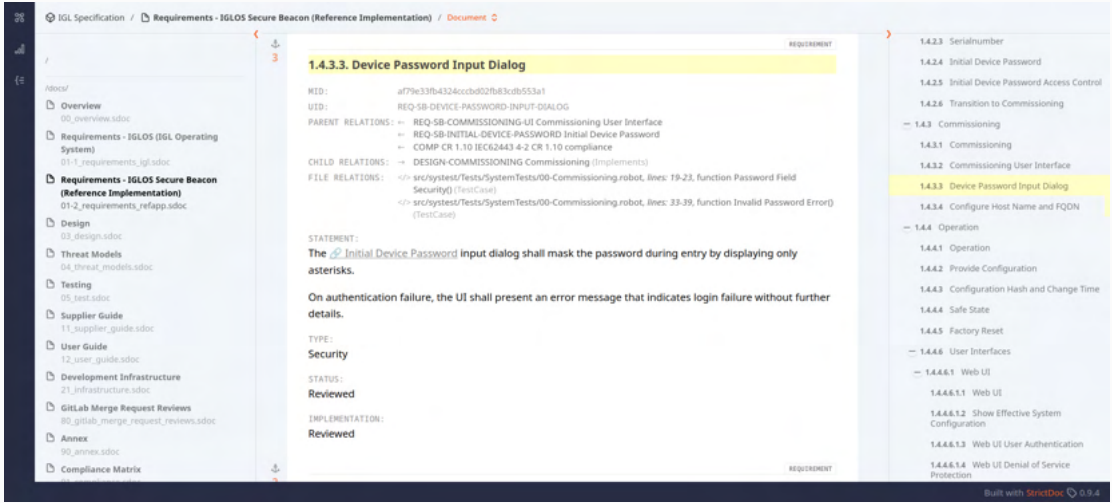
StrictDoc Format

```
[REQUIREMENT]
MID: af79e33fb4324cccbd02fb83cdb553a1
UID: REQ-SB-DEVICE-PASSWORD-INPUT-DIALOG
TITLE: Device Password Input Dialog
TYPE: Security
STATUS: Reviewed
IMPLEMENTATION: Reviewed
STATEMENT: >>>
The [LINK: REQ-SB-INITIAL-DEVICE-PASSWORD] input dialog shall mask the password
during entry by displaying only asterisks.

On authentication failure, the UI shall present an error message that indicates
login failure without further details.
<<<
RELATIONS:
- TYPE: Parent
  VALUE: REQ-SB-COMMISSIONING-UI
- TYPE: Parent
  VALUE: REQ-SB-INITIAL-DEVICE-PASSWORD
```

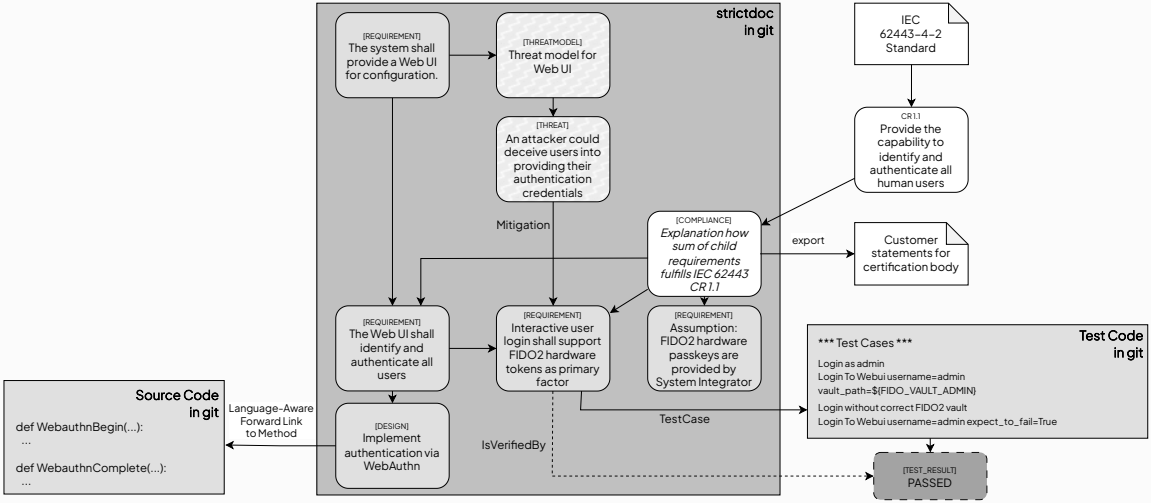
Open Source for Traceable Documentation

LINUTRONIX



Open Source for Traceable Documentation

LINUTRONIX



LINUTRONIX

How is Open Source Relevant for a Product Certification?

Traceable Documentation

Technical Requirements

External Components Management

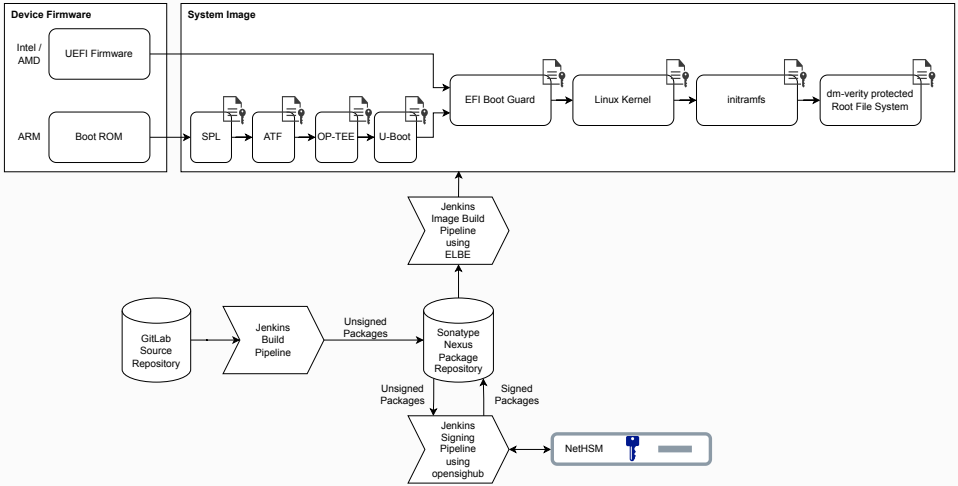
LINUTRONIX

IEC 62443-4-2 EDR 3.14 - Integrity of the boot process

Embedded devices shall verify the integrity of the firmware, software, and configuration data needed for the component's boot and runtime processes prior to use.

LINUTRONIX

IGLOS Secure Boot Chain and Signing Infrastructure



LINUTRONIX

How is Open Source Relevant for a Product Certification?

Traceable Documentation

Technical Requirements

External Components Management

LINUTRONIX

IEC 62443-4-1 SM-9: Security requirements for externally provided component

A process shall be employed to identify and manage the security risks of all externally provided components used within the product. [...]

LINUTRONIX

XZ Utils Backdoor

2005–2008 — Lasse Collin et al. develop the xz file compression

2021–2024 — Attacker uses social engineering to gain maintainer permissions

> **Pressure & gaining trust over years**
> Innocent patches to gain trust *and* persistent pressure on Lasse Collin (who mentions limited resources and mental health issues)

2024-02-26 — Disguised backdoor uploaded to Debian unstable (never stable!)

2024-03-28 — Version reverted after Andres Freund notices ≈ 500 ms longer login times

> **Days to detect, hours to fix**
> Broad community and access to source code enabled rapid response

Source: https://research.swtch.com/xz-timeline

LINUTRONIX

Better Use Proprietary Software?

SolarWinds Orion
Proprietary software for IT performance monitoring
Around February 2020: Attackers inject backdoor in official builds
Many large and governmental organizations highly vulnerable
About 10 months to find & publicly disclose

Realtek Jungle SDK
SDK for building network devices
CVE-2021-35392 - CVE-2021-35395, including remote code execution
Millions of attacks by botnets
Still many vulnerable devices expected today

LINUTRONIX

Better Develop Everything Yourself?

📱 WhatsApp

Remote code execution via custom MP4 parsing
CVE-2019-11931

💿 DVD Content Scramble System

Custom algorithm and implementation → hacked

🎮 PlayStation 3

Custom ECDSA implementation → hacked

🚗 Keyless Car Entry

Multiple cases of weak custom crypto

LINUTRONIX

How to Select Software Components?

	Security vulnerabilities	Independently auditable & patchable	Supplier can be held liable
Proprietary Software	😱	×	✓
Own Implementation	😱	✓	×
Unsupported Open Source	😱	✓	×
Open Source + Support	😱	✓	✓

LINUTRONIX

OpenSSF Concise Guide for Evaluating Open Source Software

Consider Necessity | Maintainer Diversity | Vulnerability Status | Testing Practices

Activity Level | Release Recency | Best Practices Certification | Security Response | Version Stability

Verify Authenticity | Dependency Management | Secure Defaults | Static Analysis | Security Audits

Repository Security | Security Scores | License Clarity | Security Documentation | Interface Stability | Suitability

Vulnerability Reporting | Security Guidance | Communication | Name Verification | Security Development | Adoption

Interface Design | Assessment Framework | Security Implementations | Security Reviews | Behavior Testing

Dependency Impact | Code Completeness | Malicious Code Check | Sandbox Testing | Test Validation

Source: https://best.openssf.org/Concise-Guide-for-Evaluating-Open-Source-Software

Even a small systems can contain hundreds of open-source components...

LINUTRONIX

Even a small systems can contain hundreds of open-source components...

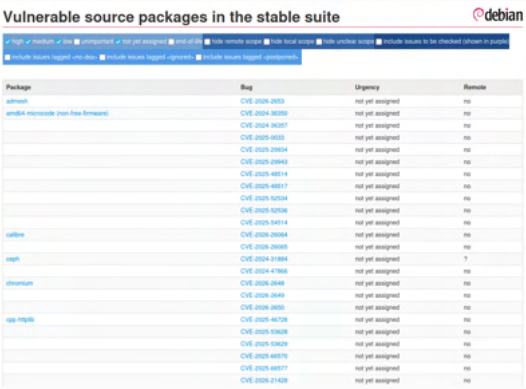
We need communities and tools!

LINUTRONIX

Security Metrics

OpenSSF Best Practices Badge

Debian Security Tracker



License Compliance

Linux Foundation project
for license scanning & compliance

OSADL project for curated
license compliance data

LINUTRONIX

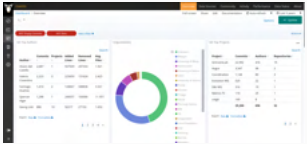
Community Health

CHAOSS

Community Health Analytics in Open Source Software

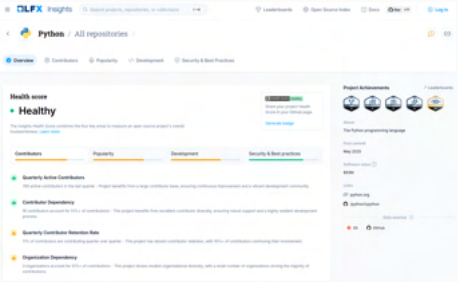
Linux Foundation project for metrics, models & software

GrimoireLab



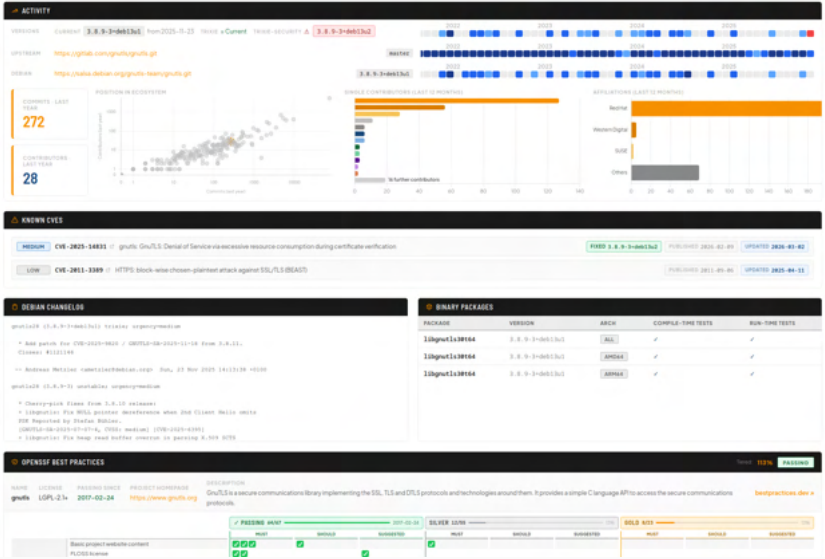
Augur

LFX | Insights



External Components Management for Open Source

LINUTRONIX



LINUTRONIX

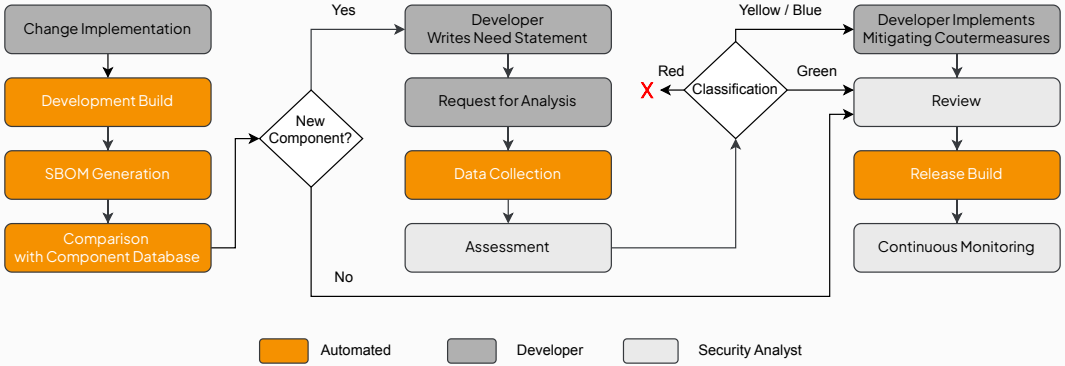
Example for an External Component Classification

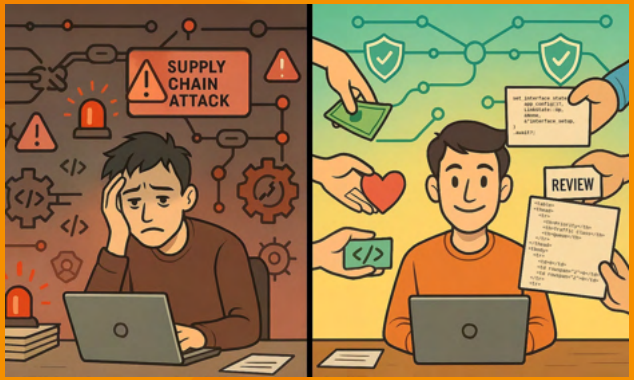
Level	Category	Example in IGLOS Context
Green	Suitable for intended use*	apache2, openssl
Yellow	Suitable for use with risk-mitigating measures*	libexpat
Blue	Suitable for limited or isolated use*	python-dbus-next
Red	Not suitable for use	OpenClaw

* with continuous monitoring

LINUTRONIX

External Component Management Process

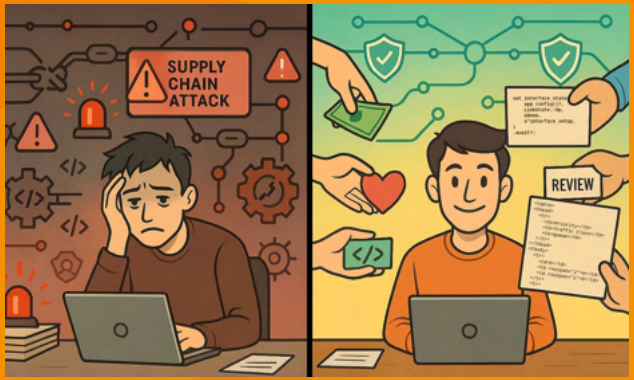




Contributing to open source and supporting maintainers is not only fair, but protects your supply chain!

LINUTRONIX

Visit
OSADL
Hall 4
Booth 4-168



Visit
Linutronix
Hall 4
Booth 4-250

Contributing to open source and supporting maintainers is not only fair, but protects your supply chain!