



Securing Embedded OSes

MICROCONTROLLERS, RTOSes AND ISOLATION KERNELS

CLOUD DAY – ESGI, PARIS – 21 OCTOBER 2026 – H2LAB

About me & H2Lab

Philippe Thierry

20 years in cybersecurity

Thales SIX OS specialist

ANSSI Embedded systems expert

Ledger Senior Staff Engineer

Safran Cybersecurity architect

H2Lab

Open hardware and open source resources for embedded systems.

Hardware platforms, analysis tools and technical documentation.

Security implementation guides (STIGs), used as references in this talk.

blog.h2lab.org

Scope, devices and learning objectives

Devices and assets

Military: tactical radio, mission computer, drone; secrets and commands.

Transport: avionics, railway signalling, automotive ECU; measurements and actions.

Industry / OT: PLC, protection relay, refrigeration; process safety and continuity.

Sensitive devices: voting machine, meter, payment terminal; results, identity and traceability.

Route: threats → runtime isolation and evidence → lifecycle trust → keys and hardware.

Scope

MCUs without an MMU, native firmware, RTOSes, microkernels. **Strict bare metal:** no OS.

Here: also OSes running directly on hardware.

1. Contain compromise.
2. Maintain lifecycle trust.
3. Link guarantees to evidence.

Why not just patch it?

Hardware and timing budgets

Limited RAM/flash; constrained MPU; sometimes no protection. Interrupts, DMA, energy and real-time deadlines.

Long service life

Heterogeneous fleets, infrequent maintenance, obsolete components, algorithms and keys requiring renewal.

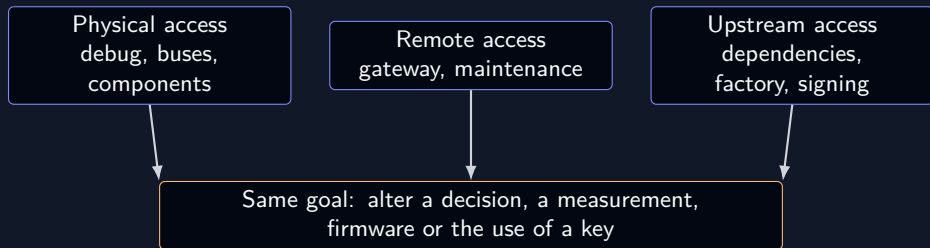
Cost of change

Impact analysis, regression testing, assurance evidence, requalification depending on the scope of change.

Distinguish the assurance frameworks

DO-178C / DO-326A in avionics; IEC 62443 in industry. Safety compliance is not proof of cybersecurity.

Threats: more connected devices, more entry paths



Sources: CISA, Volt Typhoon, 2024 ; H2Lab STIG, security components.

Avionics buses and refrigeration: the input boundary

ARINC 429: research demonstration

Trask et al., 2024: denial of service against a multifunction display on a hardware-in-the-loop testbed. No native cryptographic bus authentication.

Testbed demonstration $\neq$ aircraft compromised in flight.

Retail refrigeration: France, 11–12 October 2026

Refrigeration control altered across several retail chains; consequences for equipment and food.

Danfoss confirms incidents, including on obsolete systems; the exact technical attack vector remains to be established.

Sources: Trask et al., ARINC 429, 2024 ; franceinfo, retailer and Danfoss statements, 12 October 2026.

TRITON: targeting the safety function itself



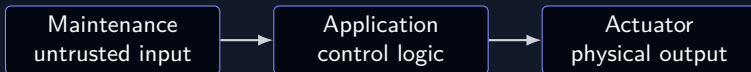
- 2017: documented industrial incident targeting a safety instrumented system (SIS).
- Distinguish process-control PLCs from safety instrumented systems.
- Relevant boundaries: programming, firmware, memory, operating mode.

Key takeaways

Assume an entry point is compromised: validate and bound hostile input, isolate critical functions, and distinguish documented incidents from hypotheses. Network controls alone cannot contain every failure.

Sources: [CISA](#), [Triconex](#) / vulnerabilities targeted by HatMan, 2018.

Industrial controller: four principles to contain compromise

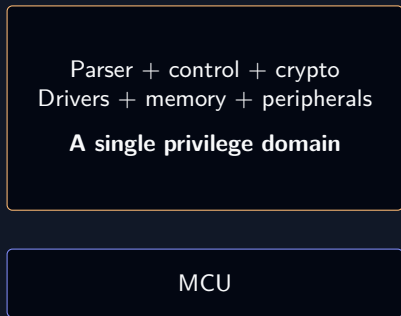


Invariant: hostile input must not access keys, write the output directly or cause a critical deadline to be missed.

- **Defence in depth:** boot, isolation, policy.
- **Task separation:** memory domains, userspace drivers.
- **Least privilege:** authorised resources and IPC.
- **Monitoring:** faults, quotas, watchdog, protected events.

From shared memory to trust domains

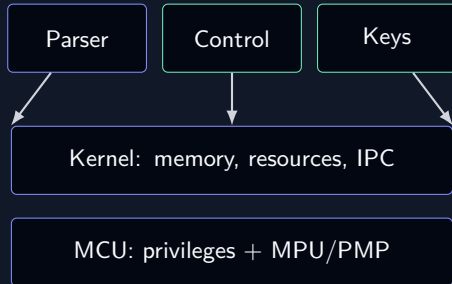
Superloop / RTOS without isolation



A memory overrun can affect every function.

Sources: [Sentry](#), [microkernel model](#). Educational diagram.

Isolation kernel



TCB: minimal trusted computing base, not just the kernel.

MPU / PMP: enforce memory boundaries, not just draw them

MPU: Memory Protection Unit; PMP: Physical Memory Protection. These enforce the memory domains shown on the previous slide.

- Unprivileged tasks; configuration restricted to the kernel.
- Read/execute code; read/write data: **W $\oplus$ X**.
- Assigned peripherals; bounded stacks and shared regions.
- Context switches: restore permissions without leaking access.

Blind spots

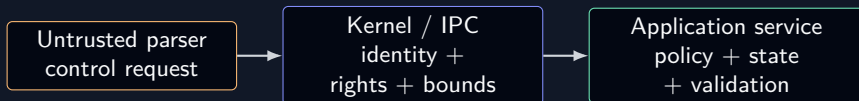
DMA and other bus masters; region count/alignment; interrupts; shared buffers; memory aliases.

The CPU's MPU does not necessarily filter DMA accesses.

Sources: H2Lab MPU STIG: ARMv7-M, ARMv8-M, PowerPC e200, RISC-V PMP.

IPC and capabilities: a pointer is not an authorisation

Isolated tasks still need to cooperate: every crossing of a boundary must be authorised.



- Explicit resources: device, interrupt, DMA, memory, service.
- Do not treat a caller-supplied identity as authoritative.
- Bound messages and queues; handle revocation and mode transitions.

Sources: [Sentry](#): declared capabilities and resources.

Spatial separation does not guarantee deadlines

Execution budget

Priorities, quotas, ISR duration, contention and IPC costs.
Measure the relevant worst case, not just the average.

Independent monitoring

Watchdog, fault counters, blocked-task detection,
bounded and actionable events.

Example

An isolated network task floods the system with interrupts.

It cannot read the keys, but the control loop misses its deadline.

Response: bound the workload, isolate resources, define safe degraded behaviour.

Demonstrating robustness: evidence, not slogans

We have defined boundaries and budgets. How do we establish that they hold?

Method	Contribution	Limits to state
Tests / fuzzing	Concrete bugs; protocols and sequences	Explored cases only
Frama-C Eva	No runtime errors within the model	Covered inputs, stubs and code
Frama-C WP	Functional contracts / ACSL invariants	Assumptions, unproved obligations
BINSEC	Binary-level properties	ISA, memory model, exploration
Physical evaluation	Resistance of the actual device	Attacker, resources and scope

Sources: [Eva](#) ; [WP](#) ; [BINSEC](#).

WooKey / ANSSI: architecture, methods and evaluation

Concrete example

STM32-based encrypted USB storage, authentication token, secure updates, open hardware and software.

- EwoK: microkernel, memory and resource isolation.
- Ada/SPARK and formal methods for sensitive modules.
- $W \oplus X$ mappings documented as proved.

Lessons learned

Assurance comes from architecture **and** implementation.

2019 inter-CESTI challenge, SSTIC 2020 feedback: also evaluate physical and software attack paths.

Sources: WooKey, SSTIC 2018 ; EwoK ; Inter-CESTI, SSTIC 2020.

Frama-C: from C contracts to proof obligations

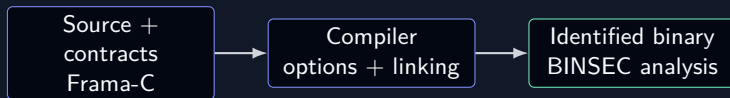
```
/*@ assigns \nothing;  
   ensures \result <==>  
     (off <= cap && len <= cap - off);  
*/  
int fits(unsigned off, unsigned len,  
         unsigned cap)  
{  
    return off <= cap && len <= cap - off;  
}
```

Educational example: check a range *without* computing `off + len`, which may overflow.

Sources: Eva ; WP ; Sentry: MPU proof examples.

- **Eva**: value domains; runtime-error alarms.
- **WP**: ACSL contracts, preconditions and postconditions.
- **Sentry**: published MPU and $W \oplus X$ invariant examples.
- In **CI**: retain code, options, models, obligations and results.

BINSEC: verify what will actually execute



- Binary analysis: symbolic execution, abstract interpretation, SMT solving, depending on the module.
- Published case: **BINSEC/REL**, cryptographic selection `ct_select`; some Clang optimisations cause the binary to lose its constant-time property.
- Compare two executions with different secrets: identical observable branches and memory addresses? This model does not cover every physical side channel.

Key takeaways

Containment depends on enforced memory and resource boundaries, authorised IPC and bounded execution; verify each property against the delivered binary, with its scope and assumptions stated.

Sources: [BINSEC: results and artifacts](#) ; [BINSEC/REL](#), IEEE S&P 2020, Table IV.

Secure boot: a chain, not a standalone check

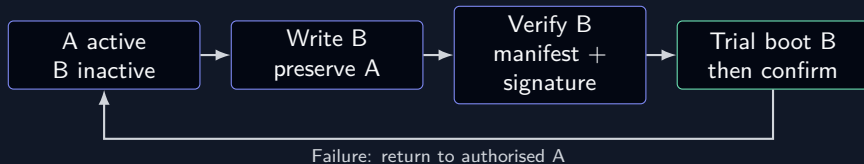


- Authenticate before execution; check target, version and dependencies.
- Protect trust anchors, the boot decision and anti-rollback state.
- Include recovery, debug and critical configuration.

Sources: H2Lab physical protection STIG ; NIST SP 800-193.

A/B updates: authenticity and power-loss resilience

Secure boot checks the installed image. Updating must preserve that trust while replacing it.



- Robust bank-selection metadata; bounded trials; controlled recovery.
- Anti-rollback: reject versions below the security floor.
- Raise that floor only with a viable recovery strategy.

Sources: H2Lab flip-flop update STIG.

Anti-tamper: prevent, detect, respond

Boot and update checks also need protection against physical bypass.

Prevent / delay

Locked debug, controlled test modes, secret protection, suitable board layout and enclosure.

Detect

Enclosure opening, abnormal voltage/clock, faults; detectors and traces depending on the hardware.

Respond

Block secret use, isolate a function, alert; erase only when appropriate and reliable.

Protect the enforcement path

A locked enclosure is not enough if debug or a fault can bypass boot verification or expose secrets.

Sources: H2Lab JTAG/SWD STIG ; H2Lab physical protection STIG.

Supply chain: from the Ledger incident to device delivery

Ledger Connect Kit – 14 December 2023

Unrevoked npm access + phishing → malicious library → deceived user signs a transaction. **No demonstrated key extraction from a secure element (SE).**

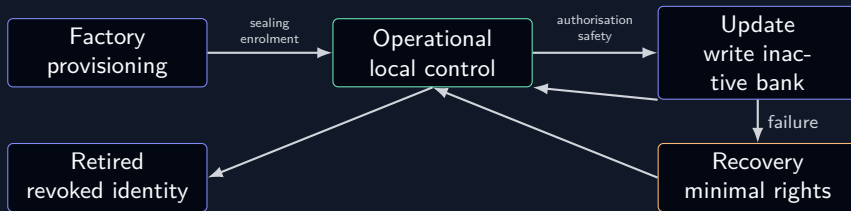


- Control dependencies and publishing access; revoke obsolete rights.
- SBOM, provenance, binary hash, reports tied to a revision.
- Separate development, release signing and identity provisioning.
- Verify factory states: debug, test keys, locks, version.

Sources: [Ledger incident report](#), 20 December 2023 ; [H2Lab C development STIG](#) ; [H2Lab security components STIG](#).

Modes and roles: a security state machine

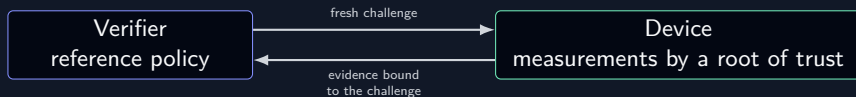
Those responsibilities become explicit device states and transition guards.



- Operator: process operation; maintainer: bounded intervention; release authority: firmware approval.
- Guards: identity, role, local presence if required, process state, admissible image; log transitions.

Attestation: prove what, to whom, and at what time?

Local checks enforce a state. Attestation lets a remote verifier evaluate evidence about that state.



- Identity, firmware/configuration measurements, mode, version and freshness.
- The attestation key signs **trustworthy** measurements, not values chosen by a compromised task.
- The verifier checks a policy and decides whether to admit the device.

Key takeaways

Maintain trust across boot, update, recovery and operation: authenticate versions, preserve an admissible recovery path, and base attestation on fresh measurements from a protected root of trust.

Sources: [RFC 9334](#), [RATS architecture](#).

Keys have a lifecycle too

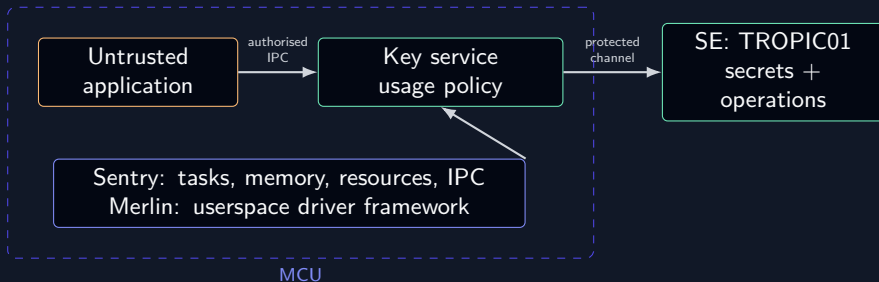
Boot, updates and attestation rely on keys with different owners and purposes.

Stage	Decision / mechanism	Typical pitfall
Generate	Qualified entropy source, health tests, DRBG	Cloned identities, predictable seed
Provision	Unique identity, authenticated channel and actor	One key shared across the fleet
Use	Purpose, role, mode and quotas; non-exported secrets	Signing oracle accessible to everyone
Renew	Rotation, controlled overlap, revocation	Offline devices lose access
Retire	Revoke identity, destroy secrets as the medium allows	Secrets persist after decommissioning

Sources: H2Lab security components STIG ; NIST SP 800-57.

Camelot / Sentry + SE: who protects which boundary?

In our controller: isolate the key service and connect it to a secure element.



OS: isolate clients and driver; service: authorise requests; SE: protect keys and primitives according to its configuration.

Sources: Sentry ; Merlin ; libtropic SDK.

Tropic Square: open hardware enables auditing

The OS–SE boundary needs evidence too. What can we inspect in the hardware component?

TROPIC01

SE built around a RISC-V core and the SPECT crypto coprocessor. RTL, application firmware, architecture and reports are available.

- Independent review and reproducible analysis.
- Public reports: entropy, side channels, fault injection.

Open is not absolute

The bootloader is not published because of third-party IP restrictions.

Check part number, silicon revision, errata and compatibility: SDK / firmware / SPECT.

Key takeaways

Protect not only key material but its use: separate key purposes, restrict crypto services and verify every OS–SE boundary. Openness enables scrutiny; it is not certification or invulnerability.

Sources: TROPIC01: [design and limits](#) ; [security reports](#) ; [SDK/firmware compatibility](#).

Exercise: maintenance is compromised. What remains protected?

Scenario

Return to our industrial controller: its maintenance parser executes hostile code. A new image arrives and power fails while it is being written. The controller has an SE and a critical physical output.

1. **Isolation:** which writes and key operations must be impossible?
2. **Authority:** what conditions authorise entry into update mode?
3. **Continuity:** which image may boot after the power failure?
4. **Assurance:** what evidence allows the supervisor to readmit the device?
5. **Consistency:** what protection survives another link's failure?

Conclusion: secure the whole chain, not just the kernel



- **Contain compromise** with isolation and least privilege.
- **Maintain trust** from boot to update and retirement.
- **Demonstrate guarantees** on the delivered implementation.

A protected key does not imply a protected decision.

The seven H2Lab STIGs

WooKey / ANSSI

Camelot / Sentry

BINSEC and Frama-C

Tropic Square / TROPIC01

The seven H2Lab STIGs: runtime and implementation

Physical memory protection – MPU-001

Threat model, regions and rights, MPU/DMA limits, negative tests.

[Guide overview](#) [English PDF](#)

Secure embedded C development – EMDEV-001

Contracts, portability, absence of runtime errors, SBOM and delivered artifacts.

[Guide overview](#) [English PDF](#)

Security component selection – SECCOMP-001

SE/TPM/TEE/HSE selection, provisioning, integration, operation and audit.

[Guide overview](#) [English PDF](#)

Microarchitectural attacks – MAA-001

Processor-level threats, applicability analysis and mitigations for embedded systems.

[Guide overview](#) [English PDF](#)

Sources: H2Lab catalogue consulted for this lecture: seven listed guides; identifiers taken from filenames.

The seven H2Lab STIGs: maintenance and physical access

Secure flip-flop updates – MAJ-001

Dual banks, verification, activation, confirmation and power-loss recovery.

[Guide overview](#) [English PDF](#)

JTAG/SWD security – JTAG-001

Maintenance access, fuses and locks, production and traceability.

[Guide overview](#) [English PDF](#)

Physical protection mechanisms – PHY-001

Enclosure opening, components, interfaces, boot and updates; attacker profiles.

[Guide overview](#) [English PDF](#)

Sources: [H2Lab catalogue](#); consult the guide revision applicable to the product.

References: threats and status of the examples

- **TRITON / HatMan, 2017** : [CISA advisory ICSA-18-107-02](#), published in 2018. 2017 industrial incident; safety controllers.
- **Volt Typhoon, 2024** : [joint advisory AA24-038A](#). Pre-positioning in critical infrastructure; not evidence of an MCU kernel vulnerability.
- **ARINC 429, 2024** : [Trask et al., HIL testbed](#). Demonstrated DoS on a multifunction display; presentation to Boeing unverified.
- **Refrigeration, France, 11–12 October 2026**: [franceinfo](#), [Danfoss and retailer statements](#). Publicly confirmed incident; technical vector not established here.
- **Ledger Connect Kit, 14 December 2023**: [vendor report](#). Software supply chain; no evidence of key extraction from the SE.

References: architectures and formal verification

- WooKey: USB devices strike back! – SSTIC 2018 ; EwoK architecture.
- Inter-CESTI challenge – SSTIC 2020: physical and software evaluation.
- Sentry documentation ; Merlin driver framework ; Sentry: formal verification of MPU invariants.
- Frama-C Eva ; WP ; ACSL contract language.
- BINSEC results ; BINSEC/REL: constant time and compilation effects, IEEE S&P 2020 ; evaluation benchmarks.

References: SEs, system consistency and lifecycle

- TROPIC01: product, RTL and firmware ; security reports ; libtropic SDK and compatibility.
- From dusk till dawn: toward an effective trusted UI – SSTIC 2023.
- Once upon a time in IoT: an industry-grade OS perspective – SSTIC 2024.
- NIST SP 800-193 : firmware protection, detection and recovery.
- NIST SP 800-57, Part 1, Rev. 5: key management.
- RFC 9334 : RATS attestation roles, evidence and policies.

Acceptance checklist: verify the claimed properties

Property	Test / analysis	Retained evidence
Isolation	Forbidden writes, MMIO, DMA, context switches	Policy + denial traces
Bounded inputs	Boundary values, fuzzing, Eva/WP within scope	Contracts + addressed alarms
Real time	IRQ/IPC flooding, contention and failures	Target measurements + assumptions
Boot / update	Wrong target/signature/version; power loss at each stage	Test matrix + flash state
Key usage	Forbidden client, wrong mode, replay, revocation	SE/service policy + traces
Delivery	Link source, build, proof, signature, provisioning	Hash + SBOM + batch records