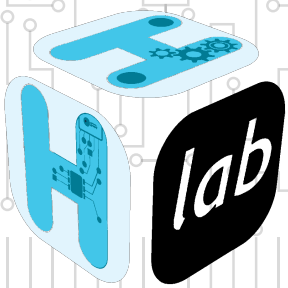
H2LAB-STIG-PHY
05-2026

Embedded systems: Guide to physical protection mechanisms



STIG H²Lab

Information

Created in 2020, H2Lab is a non-profit organization dedicated to designing, developing, and sharing open-hardware and open-source solutions for experimentation around embedded systems. Its main areas of work are:

- Design of hardware platforms for hardware and software analysis
- Development of open alternatives to proprietary, often opaque tools
- Publication of technical guides and recommendations to promote bestpractices in security, privacy, and sustainability across the ecosystemof connected devices and embedded systems

Support for researchers, teachers, and students through the provision of tools, educational content, and training.
Academic partnerships to encourage sharing of resources and knowledge.

Revision history

Version	Date	Author	Changes
1.0	May 2026	H2Lab	Initial version of the guide
1.1	August 2026	H2Lab	Formatting updates, addition of the license

Contents

Glossary of acronyms

0.1 Glossary

Side-channel attack
: Information extraction technique exploiting indirect physical measurements (power consumption, electromagnetic radiation, execution time) without altering the component's behavior.

Fault attack
: Technique consisting of deliberately disturbing the power supply, clock or environment of a component to trigger an exploitable abnormal behavior.

Semi-invasive attack
: Class of attacks requiring partial opening of the package (decapsulation) without direct electrical alteration of the circuit, as opposed to invasive attacks.

BootROM
: Read-only memory internal to the SoC executed first at boot time, generally non-modifiable after manufacturing.

Trust chain
: Ordered set of cryptographic verification steps linking the root of trust to the final application software.

Defense in depth
: Security principle consisting of layering several independent protection mechanisms so that an isolated failure does not compromise the whole system.

Secure element
: Hardware component dedicated to storing and handling cryptographic secrets, designed to resist physical and logical attacks.

Trace burial
: PCB routing technique consisting of placing sensitive traces on inner layers to limit their physical accessibility.

Micro-probing
: Attack technique consisting of directly contacting a trace or test point with a fine probe to observe or inject a signal.

DFU mode / factory mode
: Privileged operating mode intended for update or production purposes, offering extended capabilities that must be strictly controlled in operation.

Anti-rollback policy
: Mechanism preventing the installation of a firmware version older than

a minimum authorized version, in order to prevent the reintroduction of fixed vulnerabilities.

Root of trust Minimal set of code and cryptographic keys, resistant to modification, on which the entire boot verification chain relies.

Reballing Operation of rebuilding the solder balls of a BGA package, used notably during a rework or an attack requiring removal of the component.

Rework Set of operations of removing, modifying and replacing components on a PCB, which can be diverted for offensive purposes (interposition, substitution).

Secure Boot Secure boot mechanism verifying the authenticity and integrity of each software stage before its execution.

TEE (trusted execution environment) Execution zone isolated from the main operating system, guaranteeing the confidentiality and integrity of the code and data processed within it.

Plugged/tented vias PCB manufacturing techniques making transition vias inaccessible from the surface, limiting attachment points for micro-probing.

0.2 List of acronyms

AHB Advanced High-performance Bus.

APB Advanced Peripheral Bus.

API Application Programming Interface.

AVA_VAN Common Criteria attack potential evaluation class.

BIOS Basic Input/Output System.

BGA Ball Grid Array.

CC Common Criteria.

CERT Computer Emergency Response Team.

CISA Cybersecurity and Infrastructure Security Agency.

CPU Central Processing Unit.

CVE Common Vulnerabilities and Exposures.

CWE Common Weakness Enumeration.

DFU Device Firmware Update.

DSP Digital Signal Processor.

EEPROM Electrically Erasable Programmable Read-Only Memory.

FPGA Field-Programmable Gate Array.

FSBL First Stage Boot Loader.

GPIO General Purpose Input/Output.

I2C Inter-Integrated Circuit.

ICS Industrial Control Systems.

IP Intellectual Property (integrated hardware block).

ISMS Information Security Management System.

HSE Hardware Security Engine.

JTAG Joint Test Action Group (IEEE 1149.x).

MAC Media Access Control.

MITM Man-In-The-Middle.

MCO/MCS Operational condition maintenance / security condition maintenance.

MMU Memory Management Unit.

NIST National Institute of Standards and Technology.

NOR/NAND Non-volatile flash memory technologies.

PCB Printed Circuit Board.

PHY Physical communication layer.

RAM/SRAM Random Access Memory / Static RAM.

ROM Read-Only Memory.

RTOS Real-Time Operating System.

SoC System on Chip.

SPI Serial Peripheral Interface.

SSBL Second Stage Boot Loader.

TPM Trusted Platform Module.

TEE Trusted Execution Environment.

TrustZone ARM secure partitioning technology.

UART Universal Asynchronous Receiver-Transmitter.

USB Universal Serial Bus.

1
Introduction and scope

1.1 Purpose of the guide

This guide formalizes a set of technical requirements applicable to the design and cybersecurity assessment of embedded systems. The document covers the hardware and software chain, from physical access to secure boot mechanisms.

1.2 Technical scope

The scope covers:

- embedded equipment based on SoC, microcontroller or FPGA;
- physical and logical interfaces accessible in a laboratory;
- physical protection countermeasures (seals, sensors, erasure);
- software trust mechanisms (secure boot, TEE, operational/security maintenance).

1.3 Security levels and threats

The guide uses the standard attacker levels from the Common Criteria (AVA_VAN family, Common Criteria Part 3 [10]):

- **Basic**: limited capabilities and resources;
- **Enhanced-Basic**: equipped attacker with increased effort;
- **Moderate**: confirmed skills and laboratory resources;
- **High**: expert attacker, advanced resources and extended time.

  In this guide, each requirement is associated with a minimum expected AVA_VAN level. The reference threats are:

- **T01** unauthorized opening of the enclosure;

- **T02** extraction or replacement of components;
- **T03** abuse of debug and test interfaces;
- **T04** compromise of update or boot;
- **T05** information leakage via bus or external memory.

1.4 Operational summary

1.4.1 Guiding principles

- Design the equipment with a full life-cycle vision (operational/security maintenance).
- Lock down test interfaces as early as the industrialization phase.
- Apply defense in depth: passive + active + software verification.
- Measure and trace: without proof of verification, there is no actionable security.

1.4.2 Prioritization of actions

1. Establish the hardware/software mapping and asset criticality.
2. Address uncontrolled physical access and debug pathways as a priority.
3. Harden the boot chain and update mechanism before functional extension.
4. Implement periodic verification of self-protection mechanisms.

1.5 Diagram 1 – Physical attack surface

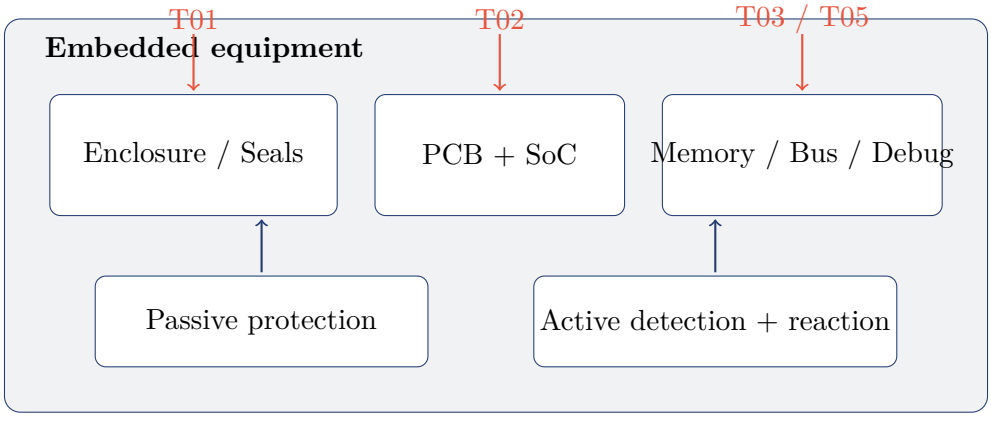


Figure 1.1: Summary view of physical exposure and countermeasures.

2
Embedded systems ecosystem

2.1 Diversity of targets

The ecosystem covers consumer products (USB key, smartphone), industrial (PLCs, rail), transport (avionics) and access control equipment. The direct consequence is a strong variability in power, maintenance and lifetime constraints.

2.2 Reference architecture

Modern platforms mostly rely on SoCs integrating CPU, memory controllers, accelerators and input/output interfaces. External connectivity is provided via GPIO and dedicated controllers, while internal buses (AHB/APB for example) remain invisible outside the chip.

REQ_ARCH_01 – Architecture mapping

The manufacturer must produce a maintained mapping of the product: SoC, volatile and non-volatile memories, coprocessors, exposed interfaces, test points and boot paths. The mapping is version-controlled with each hardware and software change.

- Minimum attacker level: **AVA_VAN.Basic**.
- Targeted threat: **T02** and **T03**.
- Without a reliable technical inventory, the real attack surfaces are not under control and subsequent analyses are not reproducible.
- References: NIST SP 800-193 (firmware resilience), BSI IT-Grundschutz Compendium (threat modeling) [23, 8].

State of the art of attacks. Hardware reverse engineering campaigns primarily exploit the absence of mapping to quickly identify critical buses, memories and test points [25, 11].

3
Technical analysis approach

3.1 Analysis sequence

The approach is conducted in five steps:

1. information gathering (documentation, certifications, patents, field feedback);
2. acquisition of several samples of the target;
3. opening and physical characterization of the PCB;
4. identification of components and interfaces;
5. passive then active interfacing, with test traceability.

REQ_METH_02 – Reproducibility of analyses

Each analysis step must be documented in a technical logbook: hardware version analyzed, tools used, test conditions, results obtained and observed limitations. Captures (bus, dumps, photographs) are timestamped and attached to the sample.

- Minimum attacker level: **AVA_VAN.Basic**.
- Targeted threat: **T03** and **T05**.
- Reproducibility is necessary to distinguish a structural weakness from a manipulation artifact and to drive corrections.
- References: CERT/CC Vulnerability Notes (documentation methodology), CERT-FR hardening recommendations [9, 2].

State of the art of attacks. Firmware reverse-engineering work shows that a non-reproducible test protocol hides the real injection vectors and delays the correction of flaws [11, 24].

4

Protection against unauthorized physical access

4.1 Study of self-protection solutions

The mechanisms observed in the ANSSI reference material fall into three families:

- **Seals and passive devices**: labels, plastic seals, metal seals, proprietary screws, structural adhesives;
- **Active detection**: opening, light and proximity sensors, conductive loop and alarm logic;
- **Active reaction**: SRAM erasure, secret invalidation, blocking of critical functions, switching to degraded mode.

4.2 Limits of passive protections

Passive seals delay and reveal opening, but can be bypassed (solvents, temperature, replacement, 3D printing). They remain relevant for controlled maintenance contexts, but are not sufficient for critical assets.

REQ_PHY_10 – Minimum passive device

Every device must combine at least two different passive mechanisms: an anti-opening mechanism (screw/seal) and an anti-substitution mechanism (unique identifier, tamper-evident marking or equivalent). Maintenance procedures must systematically check these indicators.

- Minimum attacker level: **AVA_VAN.Enhanced-Basic**.
- Targeted threat: **T01**.
- Defense in depth reduces simple bypasses and increases the probability of detecting a physical compromise.
- References: BSI IT-Grundschutz (organizational and technical measures), BSI 200-1/200-2/200-3 framework and reported solvent/heat physical attacks observed in embedded contexts [8, 5, 6, 7].

State of the art of attacks. The literature on tamper resistance and physical substitution shows that a single passive mechanism can be bypassed with standard laboratory means [1, 25].

4.3 PCB hardening against local physical access

The routing and placement of interfaces on the board directly influence the cost of a hardware attack. Three complementary levers are recommended: burying critical traces, dispersing physical access points (connectors and test points), and preferring packages that make direct probing more costly on critical components.

REQ_PHY_11 – Burying sensitive traces

Traces carrying secrets, secure boot signals or administration buses must be routed mostly on inner layers, with strict limitation of segments accessible on the outer layer. Transition vias for these signals must be protected (microvias, plugged/tented vias or equivalent solution) in order to reduce attachment points for micro-probing.

- Minimum attacker level: **AVA_VAN.Moderate**.
- Targeted threat: **T02** and **T05**.
- Burying traces increases the difficulty of locating and instrumenting critical buses, and lengthens the time required for the attack.
- References: IPC-2221 (multilayer routing), IPC-4761 (via protection), state of the art on physical tamper resistance [14, 13, 1].

State of the art of attacks. Semi-invasive and micro-probing attacks primarily target signals accessible on the surface; reducing observable external areas significantly increases instrumentation cost [3, 1].

REQ_PHY_12 – Dispersion of connectors and test points

Service connectors, test pads and measurement points must not form a single compact zone allowing simple simultaneous access. Their geographic dispersion on the PCB, combined with non-trivial reference signals and non-explicit markings in production, must make offensive interfacing longer and more detectable.

- Minimum attacker level: **AVA_VAN.Moderate**.
- Targeted threat: **T03** and **T05**.
- Dispersion limits the ability to quickly connect multi-signal capture and injection tools on a single setup.
- References: CWE-1191/CWE-1244 (debug exposure), reported embedded attacks on accessible interfaces [20, 21, 24].

State of the art of attacks. Hardware exploitation campaigns show that a centralized test zone (UART/JTAG/SPI grouped together) strongly reduces the entry barrier and speeds up bypasses of software protections [24, 19].

REQ_PHY_13 – Priority use of BGA for critical components

For critical components (main SoC, secret memory, secure element), the use of a **BGA** package must be preferred over exposed-lead packages (QFP/QFN/TSSOP), unless documented industrial constraints apply. If BGA is not used, a technical justification must demonstrate an equivalent level of resistance to probing and rework.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T02** and **T05**.
- BGA reduces direct signal accessibility and imposes more complex rework operations for the attack (removal, reballing, interposition).
- References: physical tamper resistance recommendations, fault/probing attack classes [1, 3].

State of the art of attacks. Attackers can bypass a BGA with advanced means, but the cost (equipment, skill, failure rate) is higher than that required for exposed-lead packages; this increased cost is sought to push the attack toward higher AVA_VAN profiles [1, 25].

4.4 Active detection and reaction

Active solutions require an autonomous power supply, a safe decision chain and periodic tests. They are a priority when secret confidentiality is critical or when the equipment operates outside permanent physical control.

REQ_PHY_20 – Active detection and secure reaction

For any critical-level asset, the product must implement: (i) electronic opening detection/sensor, (ii) generation of a non-repudiable security event, (iii) automatic reaction in less than 100 ms leading to invalidation of secret material exposed in RAM or in a secure element.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T01** and **T02**.
- A fast reaction limits the exfiltration window during a forced opening or an intrusive probe on the board.
- References: NIST SP 800-193 (protection, detection, recovery), hardware CWE classes for physical debug [23, 20, 21].

State of the art of attacks. Fault attacks and semi-invasive attacks illustrate that without fast detection/reaction, secret extraction becomes practicable in a laboratory [3, 15].

4.5 Diagram 2 – Active detection and reaction

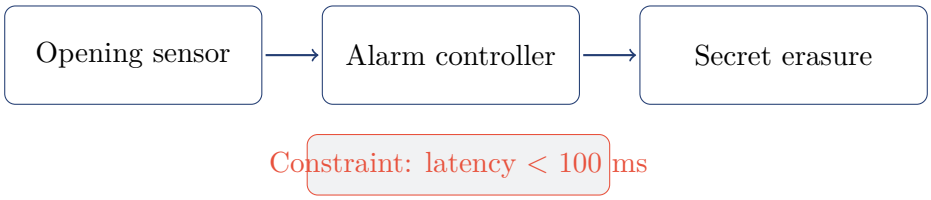


Figure 4.1: Typical sequence of active opening detection.

5

Identification of components and interfaces

5.1 Priority components

The analysis targets as a priority: serial/parallel memories, main processor, communication controllers, cryptographic coprocessors and exposed test points.

5.2 At-risk interfaces

The following interfaces must be qualified: UART, I2C, SPI, JTAG, USB/DFU, manufacturer ports, test pads, as well as service power rails that might activate hidden modes.

REQ_IF_30 – Control of debug interfaces

Debug and test interfaces in production must be disabled by default. Any re-activation must be conditioned on a strong, temporary and logged authentication mechanism.

- Minimum attacker level: **AVA_VAN.Moderate**.
- Targeted threat: **T03**.
- Field experience shows that active JTAG/UART interfaces are an immediate entry point to privileged execution.
- References: CWE-1191 and CWE-1244 (debug interfaces), example CVE-2019-18827 (secure boot bypass via early JTAG access) [20, 21, 19].

State of the art of attacks. Public demonstrations have shown that debug access in production allows secure boot bypass, memory dump and execution of unsigned code [24, 19].

6
Interfacing and attacks on embedded software

6.1 Passive and active interfacing

Passive interfacing (oscilloscope, logic analyzer, protocol decoder) makes it possible to understand exchange sequences without modifying behavior. Active interfacing (MITM, source spoofing, injection) is used to check the robustness of communication controllers and application protocols.

6.2 Update and logical surfaces

Update mechanisms, notably DFU and factory modes, constitute a major attack surface. Their exposure must be reduced and strictly controlled in operation.

REQ_SW_40 – Authenticated update chain

The update mechanism must verify signature, version and anti-rollback policy before installation. Verification keys are anchored in a non-rewritable component (fuse, secure element or equivalent).

- Minimum attacker level: **AVA_VAN.Moderate**.
- Targeted threat: **T04**.
- An unauthenticated update allows lasting introduction of a malicious firmware with persistence after reboot.
- References: NIST SP 800-147 (firmware update), NIST SP 800-193 (firmware resilience) [22, 23].

State of the art of attacks. Compromises of embedded update chains and offensive conference demonstrations show that weak control of authenticity/version opens the way to persistent takeover [16, 4, 11].

7
Root of trust and Secure Boot

7.1 End-to-end Secure Boot

Secure boot is not limited to the first loader. Integrity and authenticity verification must cover the entire chain: boot ROM, stage 1 bootloader, stage 2 bootloader, kernel, coprocessor firmware and privileged components.

Figure 7.1: Secure boot trust chain (inspired by boot stages slides).

REQ_ROOT_50 – Continuous verification of the boot chain

Each boot stage must cryptographically verify the next stage, with explicit refusal to execute on failure. Boot decisions must be logged as an event usable by supervision.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T04**.
- Breaking a single link in the trust chain cancels the benefit of secure boot and opens the way to persistent compromise.
- References: NIST SP 800-193, NIST SP 800-147 [23, 22].

State of the art of attacks. Secure boot bypasses documented at technical conferences confirm that a single unverified link is enough to execute arbitrary code as early as boot [4, 24].

7.2 TPM, HSE and immutability of the root of trust

A robust secure boot chain relies on a root of trust that is difficult to modify outside a controlled process. This root can be implemented in an immutable hardware ROM, in a TPM component, in an HSE (Hardware Security Engine) integrated into the SoC, or in a dedicated secure element depending on the equipment architecture. The goal is not technology for its own sake, but the guarantee that keys and verification decisions cannot be altered by an unauthorized update.

REQ_ROOT_60 – Immutability of the root of trust

The secure boot root of trust (anchor code, root key(s), verification policy) must be stored in an immutable mechanism or one with equivalent resistance to modification (ROM/fuse, TPM, HSE or secure element). Any change to this root must be subject to an explicitly authorized, traced and auditable key transition procedure.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T04**.
- A root that can be modified without control cancels the trust chain, even if the subsequent boot stages verify signatures.
- References: NIST SP 800-193 (roots of trust, protection/recovery) [23].

State of the art of attacks. Fault attacks and verification bypasses show that a mutable root of trust is a critical breaking point for the entire secure boot process [3, 4].

REQ_ROOT_70 – Priority use of available hardware building blocks

When the equipment integrates a security building block usable for secure boot (e.g. TPM, HSE, equivalent manufacturer mechanism), this block must be used as a priority for key anchoring, boot state attestation, or cryptographic verification of critical stages, whenever it is functionally eligible.

- Minimum attacker level: **AVA_VAN.Moderate** to **AVA_VAN.High**.
- Targeted threat: **T04** and **T05**.
- Effective use of available hardware mechanisms reduces the exposure of secrets and verification decisions in software areas that are easier to compromise.
- References: NIST SP 800-193 (device roots of trust) [23].

State of the art of attacks. Secret extraction via side channels and faults is significantly facilitated when keys remain in general-purpose components without a dedicated hardware anchor [15, 3].

REQ_ROOT_80 – Replacement technology at an equivalent level

If TPM/HSE are not available or not eligible on the equipment, the implementation must use an alternative technology offering a similar security level (for example a dedicated secure element), with documented justification of equivalence regarding key protection, resistance to modification and robustness of cryptographic operations.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T04** and **T05**.
- Using an explicit and justified alternative avoids implicit security degradation across hardware platform variations.
- References: NIST SP 800-193 and CERT/CC best practices [23, 9].

State of the art of attacks. Numerous field cases show that an uncontrolled replacement of trust mechanisms degrades resistance to physical attacks, faults and key extraction [25, 3].

REQ_ROOT_90 – Security tracking of manufacturer-resident software

The equipment manufacturer must maintain explicit security tracking of resident software supplied by component manufacturers (e.g. BootROM, security monitors, microcode and OEM boot firmware), at the same level of requirement as application software. This tracking includes: (i) CVE/advisory monitoring, (ii) evaluation of product exposure, (iii) mitigation plan when the fix cannot be deployed (workaround, configuration hardening, usage restriction, hardware replacement), (iv) operational/security maintenance traceability of decisions.

- Minimum attacker level: **AVA_VAN.High**.
- Targeted threat: **T04** and **T05**.
- Manufacturer-resident software components are attack targets just as much as application firmware; the absence of tracking creates a lasting security debt on building blocks that are sometimes unpatchable.
- References: NIST SP 800-193 (resilience and recovery), CVE cases on i.MX6 and Tegra BootROM [23, 17, 18].

State of the art of attacks. The signature verification bypass on the BootROM of the i.MX family (including i.MX6) [17] and the Fusée Gelée attack class on Tegra (publicly documented on Tegra X1) [12, 18] illustrate that manufacturer-resident software must be subject to continuous security tracking, just like application software.

Bibliography

[1] Ross Anderson and Markus Kuhn. Tamper resistance – a cautionary note. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pages 1–11. USENIX Association, 1996.

[2] ANSSI. Cert-fr: Alertes, avis et recommandations. CERT-FR Portal, 2026. Accessed 2026-05-31.

[3] Alessandro Barenghi, Luca Breveglieri, Israel Koren, and David Naccache. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proceedings of the IEEE*, 100(11):3056–3076, 2012.

[4] Yuriy Bulygin and Oleksandr Furtak. A tale of one software bypass of windows 8 secure boot. Black Hat USA, 2014.

[5] Bundesamt fuer Sicherheit in der Informationstechnik. Bsi standard 200-1: Information security management systems (isms). Technical report, BSI, 2017.

[6] Bundesamt fuer Sicherheit in der Informationstechnik. Bsi standard 200-2: It-grundschutz methodology. Technical report, BSI, 2017.

[7] Bundesamt fuer Sicherheit in der Informationstechnik. Bsi standard 200-3: Risk analysis based on it-grundschutz. Technical report, BSI, 2017.

[8] Bundesamt fuer Sicherheit in der Informationstechnik. It-grundschutz compendium. BSI Web Portal, 2026. Accessed 2026-05-31.

[9] CERT Coordination Center. Vulnerability notes database. CERT/CC Knowledge Base, 2026. Accessed 2026-05-31.

[10] Common Criteria Development Board. Common criteria for information technology security evaluation, part 3: Security assurance components. Common Criteria Portal, 2022. AVA_VAN levels: Basic, Enhanced-Basic, Moderate, High.

[11] Andrei Costin, Apostolis Zarras, and Aurelien Francillon. A large-scale analysis of the security of embedded firmwares. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 95–110, 2014.

[12] fail0verflow. Shofel2, a tegra x1 and nintendo switch exploit. Technical disclosure / exploit write-up, 2018. Attaque de type Fusée Gelée sur Tegra X1 (BootROM/RCM).

[13] IPC. Ipc-4761: Design guide for protection of printed board via structures. Industry Standard, 2017.

[14] IPC. Ipc-2221: Generic standard on printed board design. Industry Standard, 2021.

[15] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology – CRYPTO'99*, pages 388–397. Springer, 1999.

[16] Charlie Miller and Chris Valasek. Remote exploitation of an unaltered passenger vehicle. Black Hat USA, 2015.

[17] MITRE. Cve-2017-7932. Common Vulnerabilities and Exposures, 2017. NXP i.MX family: bypass de verification de signature via certificat forge.

[18] MITRE. Cve-2018-6242. Common Vulnerabilities and Exposures, 2018. NVIDIA Tegra BootROM RCM buffer overflow permettant l'execution de code non verifie.

[19] MITRE. Cve-2019-18827. Common Vulnerabilities and Exposures, 2019.

[20] MITRE. Cwe-1191: On-chip debug and test interface with improper access control. Common Weakness Enumeration, 2026. Accessed 2026-05-31.

[21] MITRE. Cwe-1244: Internal asset exposed to unsafe debug access level or state. Common Weakness Enumeration, 2026. Accessed 2026-05-31.

[22] National Institute of Standards and Technology. Bios protection guidelines. Technical Report SP 800-147, NIST, 2011.

[23] National Institute of Standards and Technology. Platform firmware resiliency guidelines. Technical Report SP 800-193, NIST, 2018.

[24] Johannes Obermaier and Florian Tatschner. Shedding too much light on a microcontroller's firmware protection. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, 2018.

[25] Mohammad Tehranipoor and Farinaz Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE Design & Test of Computers*, 27(1):10–25, 2010.

License