Technical Paper
Study on the security of connected objects

Thierry Didi,
April 2025
Tidiwi

The security of connected objects has long been set aside by manufacturers, but the current geopolitical context, as well as changes in regulations, have placed this issue at the center of the concerns of manufacturers and users.

Through my professional experiences and personal research, I have had the opportunity to explore several development frameworks created by manufacturers and non-profit organizations to ensure the security of embedded software.

This article is focusing on:

- Regulatory changes in Europe
- Existing security frameworks
- The best practices recommended by ENISA

It emphasizes the operational aspects of cybersecurity rather than the organizational aspects, without neglecting the importance of the latter:

- It is essential to develop a culture of cybersecurity within the company, involving all levels, from members of the Executive Committee to developers and buyers.
- Continuous training for employees, especially developers, is crucial to keep them informed of the latest advancements in this ever-evolving field.

I hope this article helps everyone better understand the steps needed to secure IoT devices and comply with future regulations.

If you have any comments, remarks or questions, do not hesitate to contact me by email to let me know: thierry.didi [at] tidiwi [dot] com

Introduction

Before going into detail about the possible attacks against connected objects, and the measures that can be put in place to protect yourself from them, it is useful to specify a few elements of language:

- **DDOS Attack:** (Distributed Denial Of Service) : A large number of devices generate simultaneous requests to a server to crash it.

- **Attacker:** A malicious user who seeks to disrupt the system.

- **CVE** (Common Vulnerabilities and Exposures): When a flaw is detected in open source or proprietary software, it is published in one or more public databases managed by entities officially recognized to manage this type of database (Microsoft, Google, MITRE, Apache Software Foundation, etc.). The database manager assigns it a unique identifier named CVE which has the form CVE-YYYY-NNNNN. In CVE databases, there is a more precise description of the security flaw, its impact, and possible fixes. MITRE is responsible for assigning CVEs so that the same CVE is not assigned to two different vulnerabilities.

- **Cyber Resilence Act** (CRA): Regulation that will gradually come into force in Europe to strengthen the security of equipment containing digital elements (hardware and/or software). The full text is available at https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202402847

- **RED Directive** (Radio Equipment Directive): European Directive 2014/53/EU that applies to radio equipment. The full text is available at: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014L0053

- **ENISA** (European Union Agency for Cybersecurity: https://www.enisa.europa.eu/).

  IoTSF (IOT Security Foundation: https://iotsecurityfoundation.org/) is a UK-based non-profit organization focused on the security of connected objects.

- **JTAG**: In this document, the term JTAG connector represents a connector present on electronic boards and allowing to read/write to the memory of a microcontroller, and to debug a program executed by this microcontroller. Typically, this connector is only present on boards used by developers, but its footprint is present on the PCB of all boards. It is possible (and even recommended) to disable the JTAG of processors once the products are put into production.

- **Grey Market**: this is a market where you can buy authentic products, but outside the official distribution channels authorized by the manufacturer.

- **MITRE** (https://www.mitre.org/) is a US-based non-profit organization that supports the U.S. government, particularly in the field of cybersecurity.

- **SBOM (**Software Bill Of Material): This is the list of third-party software components (often open source), which go into the composition of a software. Each software

component must be associated with its version. The advantage of knowing the SBOM of a software is to make it possible to quickly identify (and correct) the security flaws introduced by this software as soon as these flaws are published.

VEX (Vulnerability Exploitability Exchange): This is a standardized format intended to describe whether software is vulnerable to a known security vulnerability.

The Context

Manufacturers of connected objects tend to take a close interest in security when a serious incident makes the headlines (a *DDOS*[1] attack generated by connected cameras, for example, etc.), or when the implementation of a legal obligation is approaching, as is the case in Europe with the *RED directive* in 2025 or with the *Cyber Resilience Act* by the end of 2027.

The risks associated with the hacking of connected objects are often underestimated, especially when they are seemingly innocuous objects and not intended for critical environments. It's a common misconception that hacking is mostly limited to ransomware or the theft of banking and personal data. However, the motivations of attackers can be much more varied:

- It can simply be the sabotage of infrastructure, carried out by powerful states or organizations.
- Malicious competitors can launch attacks to damage the reputation of a product or company, or to "steal" another company's intellectual property.
- Isolated hackers can carry out attacks for ideological reasons or for pure entertainment.
- Some attackers may seek to exploit the connectivity and processors of a fleet of objects to conduct illegal activities, such as denial-of-service (DDoS) attacks, or spy on GPS locations or conversations.
- Attacks can also target wireless systems (NFC, radio waves) to break in, steal vehicles, or gain access to buildings by hacking into communications between a transmitter (access badge, vehicle key) and a receiver.

Finally, it should be borne in mind that hacking operations can benefit from internal complicity, with individuals within the company having access to sensitive resources.

The examples below show that hacking a mundane object can have significant consequences:

- An attacker who is able to take control of a fire detector or who manages to install a fake detector in a building could trigger false fire alarms, disrupting the operation of the business and incurring economic costs. Those erroneous warnings would also damage the reputation of the manufacturer of the fire safety system, whose reliability would be called into question. The customer could even ask for financial compensation from the manufacturer of the system.

- An attacker who is able to break into a company's access control system could significantly disrupt its operation. By creating fake access badges or cloning genuine badges, they could easily break into the company's premises.

- An attacker who can modify the software of a water, gas or electricity meter could manipulate the billing system, resulting in significant financial losses for the supplier.

- An attacker who can install new firmware on a connected object could render it completely unusable, or use it as a launch point for distributed denial-of-service (DDoS) attacks, as has already been observed. In both cases, the risks in terms of image and financial costs for the manufacturer would be considerable.

[1] Distributed Denial Of Service: A large number of devices generate concurrent requests to a server to overload it.

Frameworks

To meet the security needs of connected objects, a new regulation will come into force in Europe, concerning all objects connected to the Internet:

- An amendment to the *RED* Directive introduces new requirements for new products with wireless interfaces and placed on the market from August 2025.

- The *Cyber Resilience Act* (CRA) will gradually come into force by the end of 2027, imposing new constraints applicable in particular to connected objects. These constraints will be applicable throughout their life cycle, not just when they are placed on the market.

In addition, several organizations have specified frameworks and methodologies to improve the security of connected objects. Among them:

- *ENISA* has published a guide for manufacturers, developers and integrators to help them secure connected objects: Guidelines for Securing the Internet of Things https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Report%20-%20Guidelines%20for%20Securing%20the%20Internet%20of%20Things.pdf).

- The *IoTSF* has developed the "IOT Security Framework" dedicated to the security of connected objects: https://iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf.

- *MITRE* created the Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) database to list cybercriminal attacks and describe the techniques used. This database is accessible to all: https://attack.mitre.org/.

- *MITRE* has also developed the EMB3D (https://emb3d.mitre.org/) framework, which focuses on improving the security of embedded systems.

- ARM Limited, whose CPU cores power a large part of connected objects, has created the PSA Certified (https://www.psacertified.org/) framework for manufacturers of connected objects. This framework provides a set of design rules to ensure the security of objects. It is a very low-level framework, associated with open source modules (TF-M[2], TF-A[3]) allowing it to be implemented on ARM Cortex M or Cortex A processors.

This document focuses in particular on the guide published by *ENISA,* which undoubtedly served as the basis for the development of the *Cyber Resilience Act*.

[2] https://www.trustedfirmware.org/projects/tf-m/
[3] https://www.trustedfirmware.org/projects/tf-a/

Changes in European regulations

The two recent changes in European regulations concern an evolution of the *RED directive* [4] and the adoption of the *Cyber Resilience Act*[5].

- The new *RED* Directive will come into force in August 2025 for all new products placed on the market in Europe. It introduces three new constraints in Article 3 ("Essential Requirements"):
    - 3.3§d: *"radio equipment does not harm the network or its functioning nor misuse network resources, thereby causing an unacceptable degradation of service"*
    - 3.3§e: *"Radio equipment incorporates safeguards to ensure that the personal data and privacy of the user and of the subscriber are protected"*
    - 3.3§f: *"Radio Equipment supports certain features ensuring protection from fraud"*

  There is a family of harmonised standards to ensure compliance with these new requirements:
    - EN 18031-1:2024: Common Security Requirements for Internet Connected Radio Equipment. Used to verify compliance with RED 3.3-§d
    - EN 18031-2:2024: Common Security Requirements Equipment Processing Personal Data: Used to verify compliance with RED 3.3-§e
    - EN 18031-3:2024: Common Security Requirements for Equipment Processing Virtual Money or monetary value: Used to verify compliance with RED 3.3-§f

- The *Cyber Resilence Act* will gradually come into force by the end of 2027, imposing new constraints on electronic products, including connected objects, throughout their life cycle and not just when they are placed on the market. However, these constraints do not apply to medical devices[6] ,automotive industry[7], military devices, civil aviation, or open source software.

  As a first step, the *CRA* (Article 13) requires a risk assessment to be carried out, and the results of this assessment must be taken into account throughout all phases of the product's lifecycle, from design to market withdrawal, including maintenance operations.

  The list of requirements is easily consulted in the text of the *CRA* (Annexes I, II, and III). The most important elements are:
    - Appendix I - Part I: Product Safety
      This section concerns software updates, access control, managing the confidentiality and integrity of critical data stored in the device as well as the data transmitted, implementing measures to minimize the impact of a product malfunction on other products on the network, limiting the attack surface (UART-type interfaces, *JTAG* ... or open TCP/UDP ports), monitoring and reporting of security incidents, the ability for a user to delete all sensitive data from a device.

[4] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014L0053
[5] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202402847
[6] Cybersecurity of Medical Devices is addressed in regulations (EU) 2017/745 and (EU)2017/746
[7] Cybersecurity for the Automotive industry is addressed in regulation (EU)2019/2144

- Appendix I - Part II: Vulnerability Management
  It involves implementing processes to detect vulnerabilities, fix them, distribute patches, and make these vulnerabilities public (via *CVEs*) as soon as a patch is identified. Detected security incidents will have to be reported to *ENISA*. In particular, this management requires generating an *SBOM* (Software Bill Of Material) for the software integrated into the product. It is not mandatory to make this *SBOM* public.

The *CRA* also defines several classes of devices based on the security risk associated with each device. The majority of the devices are Class I devices. Class II corresponds to security equipment (hypervisor, firewalls). Finally, some devices are considered critical (metering gateways, smart cards). Essentially, the class of a device affects the certification process. It is possible to apply a self-certification procedure to a Class I product that implements the harmonized standards associated with the *CRA*. These harmonised standards do not yet exist (in April 2025) but they should be published well before the 2027 deadline. However, the standard "*ETSI EN 303 645 V2.1.1: Baseline Requirements"* [8] and the Technical Specification "*ETSI TS 103 701 V1.1.1: Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements"*[9] cover a large part of the requirements of Annex I of the CRA.

Most static code analysis tools (e.g. Sonarqube[10]) provide, or will provide in the near future, ways to generate an SBOM, and to list vulnerabilities introduced by third-party software.

Finally, manufacturers will be required to allow users to report any security vulnerabilities they identify, by designating a single point of contact to be notified when a vulnerability is detected.

[8] https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf
[9] https://www.etsi.org/deliver/etsi_ts/103700_103799/103701/01.01.01_60/ts_103701v010101p.pdf
[10] https://www.sonarsource.com/products/sonarqube/

The ENISA Guide

Product safety management is done throughout the life cycle of a product, from the design phase to the withdrawal from the market. The *ENISA guide* is structured around the following concepts:

- The product life cycle
- Threats
- Good practices

The Life Cycle of a Product

The concept at the heart of *ENISA*'s approach is called "Secure By Design": It is a set of principles and best practices aimed at proactively and continuously integrating security throughout the lifecycle of a product, from its design to its withdrawal from the market, in order to reduce the risk of vulnerabilities and protect against potential threats from the beginning. In concrete terms, this guide defines the following stages in the life of a product:

  - Conception
  - Development
  - Production
  - Usage
  - Support
  - Withdrawal

- **The design**
  From this stage, the security strategy must include electronics, software, packaging, the IOT platform and associated services, taking into account the intended use environment for the product.

- **Development**
  This phase includes the software development of the product, the services to be implemented in the service platform, the electronics and the packaging. Obviously, safety must be taken into account during all these phases.

- **Production**
  This step is also very important, because if the production line is not secure, there is nothing to prevent an attacker from stealing security data before the product is even on the market, or even from integrating malware into it.

- **Use and support**
  In addition, the consideration of safety does not stop when the design is finished and the product is put on the market. The manufacturer must take into account the installation and then support phases throughout the life of the product. In particular, it is important to monitor products for possible flaws or cyber-attacks, and to implement remote software update mechanisms.

- **Withdrawal**
  Finally, the phase of withdrawal from the market must be studied, for environmental reasons in the first instance, but also to prevent a product at the end of its life from being analysed to extract confidential information (personal information, encryption keys, etc.)

Threats

The most significant threats to IOT systems, as identified by experts in the field, are listed below. This section is important because it provides a better understanding of the risks that a manufacturer of connected objects will face, especially if they are not paying attention to cybersecurity:

- **Physical attacks**
  These attacks are intended to render the product non-operational, or to resell end-of-life or defective products on the *grey market*. The first point is not necessarily important if the object is not used in a critical context because the owner of the object is responsible for its physical integrity. The second point is much more important because it can lead to a loss of income for the manufacturer, or damage to its reputation if defective products are resold on the *grey market*.

- **Loss of intellectual property**
  Recovery by reverse engineering of documents, security keys, algorithms embedded in the product. These attacks can have serious legal consequences in the event of the theft of personal data, or financial consequences in the event of the theft of confidential industrial data.

- **Product cloning**
  Manufacture of fake products under the same or a different brand as the original product. Fake products can also embed malware, while masquerading as original products. Here, too, these threats can have significant financial consequences, or consequences in terms of image for the manufacturer.

- **Miscellaneous abuses**
  Since attackers have infinite time and physical access to devices, they can use advanced tools to attack them. By generating magnetic fields or by analyzing the current consumption of a device during an encryption operation, it is possible to retrieve the encryption keys used. By accessing a processor's *JTAG*, they can insert malware into the product.

- **Network attacks**
  Poorly protected devices can be used as the basis for large *DDOS* attack campaigns.

- **Attacks on the manufacturing process**
  If an attacker is able to break into the production line, he can insert malware into the products at the manufacturing stage, or recover the security keys that are supposed to protect the system.

- **Attacks at the IOT platform**
  If device authentication to the service platform is not robust enough, fake devices can connect to the platform, and transmit erroneous data, or even use the service platform as a gateway for attacks on the local network to which the platform's servers are connected.

Good practices

Finally, this guide defines a set of best practices to protect yourself from these threats. These practices can be classified under the following headings:

- Hazard identification
- Protection against risks
- Anomaly detection
- Response to abnormalities and return to normal

Best practices: identification of risks

- **Risk analysis**
  A risk analysis must be carried out at the design stage, to identify the risks involved according to the intended use of the product. For each identified risk, the consequences must be assessed, and measures must be put in place according to the severity of these consequences.

Best practices: protection against risks

- **Corporate culture**
  It is of utmost importance to develop a culture of cybersecurity, both among members of the Executive Committee and among developers or buyers of third-party software.

- **Subcontractor management**
  It is necessary to ensure the measures implemented by subcontractors to guarantee the safety of their processes. Notably, production plants must be able to guarantee that malware will not be introduced into products at the time of production, or that cryptographic elements (encryption keys, certificates, ...) will not be disclosed or tampered with during the production phase. It is also necessary to be able to ensure that the binaries that will finally be integrated into the devices are indeed those that have been provided to the company in charge of production. This last point can be achieved by inspecting a sample of products at the end of the production line, for example.

- **Generating a Software Bill Of Material (SBOM)**
  The list of plug-ins built into the product should be generated to allow for quick identification of issues associated with third-party software.

- **Factory configuration**
  When a product is restored to its "factory" configuration, security features must be enabled by default. This is because many users will not be interested in restoring the product's security settings, due to ignorance or lack of time. It is therefore important to ensure that if certain features must be disabled to perform a particular action, this disabling is the result of a conscious action by the object's operator.

- **De-commissioning**
  When a product is withdrawn from the market, for example because it is defective, it must be ensured that all confidential data (personal data, security keys, etc.) are properly erased from the product.

- **Use of third-party software**
  During development, the use of third-party software, especially open source software, introduces a significant security risk. New security vulnerabilities are regularly identified by the open source community, and patches are implemented. It is therefore always necessary to ensure that external libraries have not been modified by a third party, that the most recent versions are used, and to monitor and update these libraries throughout the life of the product.

- **The implementation of cryptography**
  The encryption algorithms themselves are regularly deprecated when the computing power of processors increases, or when security flaws are discovered. For example, the WEP protocol is banned for the security of WIFI networks, the DES or 3DES algorithms are also banned from security protocols.
  In addition, a misuse or poor implementation of a security algorithm that is recognized as very robust can introduce a security vulnerability. For example, the AES-CTR algorithm does not provide true security if the same IV (Initialization Vector) is used for multiple cryptographic operations. Similarly, AES-CBC offers no guarantee of security if the IV is not randomly selected for each cryptographic operation. It is therefore always necessary to make sure to use algorithms in line with the recommendations of security experts, and to use these algorithms according to the rules of the art.

- **The chain of trust**
  The implementation of a chain of trust (root of trust) is of primary importance for the start of the software. When a processor is started, a first software (bootloader) is executed: this software is responsible for verifying the integrity of the next software to be executed (second stage bootloader, or Operating System, etc.), which is itself responsible for verifying the integrity of the next software (client application for example). Ideally, this chain of trust is guaranteed by a security component (Secure Element) that will store the encryption keys and execute the cryptographic algorithms in a secure manner. Some recent SOCs (Systems On Chip) integrate this type of component, but it is also possible to implement these components on the PCB if the SOC does not integrate them.

- **Minimizing the object's attack surface**
  This point is one of the essential elements of the Secure By Design approach. It consists of disabling all the interfaces of the product that are not used (JTAG, UART console, TCP/UDP ports, etc.) that could serve as an entry point for an attacker.

- **Minimize the attack surface of the IOT platform**
  The services associated with the product will often be implemented within the IOT Platform. This platform will therefore be accessible to objects (often via UDP/TCP ports) so that they can transmit their data to it. It is very important to verify that this platform exposes only the services useful for the proper functioning of the system (in particular that only useful UDP/TCP ports are opened) to prevent an attacker from using services left open inadvertently (ftp, ssh, telnet, ...).

- **Access management in the IOT platform**
  Since the IOT platform is at the heart of the system, it is of paramount importance to properly manage access control there. Only properly authenticated devices should be able to connect to it, and these devices should only have access to the minimum resources they need to function, following the principle of "least privilege" which is also one of the key elements of the Secure By Design approach.

- **Using authenticated components**
  The use of authenticated components helps prevent counterfeiting. In particular, some SOC providers allow processors to be customized so that the software can only run on processors that are properly customized by the manufacturer of the connected object. That is to say, a counterfeiter who would be able to obtain an "official" binary image of an object will not be able to run it on a processor with the same reference, but that he would have bought outside the official sales channel defined by the manufacturer of the object.

- **Use of unique identifiers per product**
  Generating a unique identifier per device during production will allow that device to be authenticated to the service platform, or revoked in the event of a malfunction.

Best practices: anomaly detection

- **Remote monitoring**
  It is impossible to predict in advance all the attacks that will occur during the life of the product. It is therefore necessary to implement mechanisms for detecting abnormal situations. In particular, logs can be used to trace all security-related events that may appear during the life of the product. The continuous analysis of logs (from the service platform, and potentially thanks to AI), will make it possible to detect threats as soon as they appear by detecting abnormal behavior.

Best practices: responding to anomalies

- **Remote software update**
  It is essential to be able to remotely update the software and/or security elements (keys, certificates, etc.) to correct security flaws discovered throughout the life of the product. But you have to be absolutely sure that the update chain is not corrupted, which could allow an attacker to download malware into the connected object.

- **Fixing security vulnerabilities**
  Security vulnerabilities must be analyzed, published (as soon as possible without generating additional risk) and corrected via remote upgrades as soon as a solution has been identified. It should be noted that this mechanism will be made mandatory when the *Cyber Resilence Act* comes into force in Europe.

Conclusion

In conclusion, it is essential to secure the entire product development and production chain. An attacker will target the weakest link, which may be in a poor implementation of cryptography algorithms, insufficient protection of the manufacturing plant, or a poorly secured remote software update process.

The "Secure By Design" approach significantly reduces security risks. Increasingly used in the development of connected objects, this approach has no formal definition, but all the players who refer to it put forward the same principles. Its goal is to approach security proactively rather than reactively. This involves identifying the risks to which the device will be exposed throughout its life at the design stage, and putting in place proactive measures to detect and correct security vulnerabilities as soon as possible. Because software and threats evolve over time, this approach doesn't stop at time to market, but extends throughout the life of the product. The key elements of this approach are:

- Perform a risk analysis at the design stage to identify the threats to which the object will be exposed.
- Minimize the attack surface by disabling all unused interfaces (UART consoles, TCP/UDP ports, JTAG, etc.).
- Authenticate devices that are authorized to connect to the IoT platform to avoid fake device connections.
- Implement resource access control by applying the principle of least privilege, ensuring that each entity (connected object, user, etc.) has access only to the resources that are necessary.
- Remain vigilant with regard to third-party services, in particular external libraries (open source or proprietary) integrated into the product.
- Implement a strategy for remediating security vulnerabilities throughout the life of the product, including through software updates.
- Implement measures to detect and report intrusions, such as generating logs for anomalous behavior and analyzing those logs on an ongoing basis.
- Establish a vulnerability disclosure policy.

Finally, beyond *ENISA*'s recommendations and while awaiting the publication of harmonized standards, the guidelines published by *IoTSF* provide a detailed methodology for conducting risk analyzis and implementing practical measures to manage those risks.[11]

[11] https://iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf

ABOUT TIDIWI

TIDIWI is a consulting company specializing in the design and development of IoT projects. In particular, TIDIWI has strong expertise in wireless communication protocols and ultra-low-power systems.
TIDIWI provides consulting services, embedded software development, electronic board design, development team management, and project management.

For more information, visit TIDIWI's website: www.tidiwi.com

Or contact: thierry.didi[at]tidiwi[dot]com