VDOO Guideline

Integrating Security into the IoT SDLC

Integrating Security into the IoT SDLC

The development of an IoT device, many times referred to as an embedded system or a connected device, is a complicated task, involving many processes that are conducted by different entities. Usually, these processes are driven by several owners with different considerations, challenges, and constraints - engineers, architects, and product managers - each wants to deliver the best quality product in the fastest time to market. Once a security implementation is integrated with the other processes, things get even more complicated, and important questions need to be asked - how do we know what is the right security for the product? How do we test its security? How can we fix a security issue as fast as possible? How can we optimize the product security without slowing down the Continuous Integration (CI) process?

Integrating Security into the IoT SDLC

This article suggests a method of security integration into the IoT device development process and is designated mainly for technical professionals, yet, provides insights and benefits for those who read it from a business perspective.

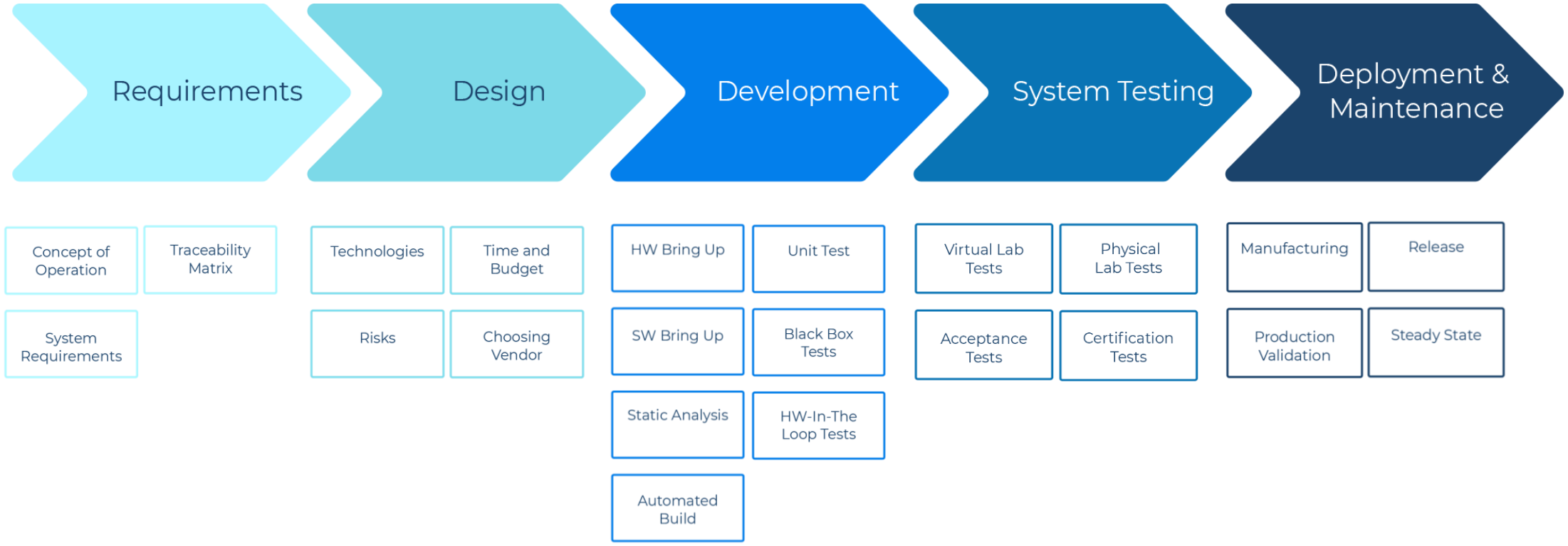
An established Software Development Life Cycle (SDLC) process, which is a framework that defines the tasks that should be performed at each step in the software development process, provides significant value related to reducing risk, ensuring business goals are met, enabling repeatable processes and more.

Below is a detailed description of each of the major phases in a typical IoT SDLC process, including the related operations and the required steps for ensuring smooth integration of the security requirements into a specific SDLC phase

Typical SDLC Phases for an IoT Product

VDOO

The number of phases in an SDLC may vary, depending on the business and on its product goals, but is typically between five and seven phases. The following diagram describes a typical track of an SDLC process for an IoT product:



Typical SDLC Phases for an IoT Product

VDOO

This typical IoT SDLC process highlights five different phases of the development process that represent a framework for technical and non-technical activities, aimed at delivering a system or a product that meets the business expectations.

In the next sections there are descriptions of each of the SDLC phases in general, and more particularly, how it is affected when the process involves security implementation for the IoT product lifecycle - from the very beginning and through all of the later phases.

Phase 1 – Requirements

VDOO

When planning a new product, the first questions that should be answered are - what's the purpose of this product? What problems should it solve? Who will use the product? How will they use it? What is the input/output of the product?

To answer these questions properly, a well-established product management process should be conducted, including the basic steps of problem definition and outline of MRD/PRD to provide a map of arguments and features as well as user stories and security implication.

Next, the **Concept of Operations** should be defined, including the characteristics of the proposed system from the user's perspective. Next, the **System Requirements Document (SRD)** should be prepared, which is the formal statement of the system requirements, including a list of functional requirements, data requirements, system interfaces, and physical requirements. Then, the **Traceability Matrix**, the table that links the requirements to their origins and traces through the project life cycle, should be prepared as well.

It is essential to define the product's security requirements at this early phase.

Phase 1 – Requirements

VDOO

To do that, first a definition of the product's key security risks is needed. For example, the type of information that the product will process, its functionality, etc.

Second, a definition of the security standards or certifications that the product should meet is needed. The decision of whether to support a specific standard usually depends on the market or industry and should be received as a requirement from the product management team.

In order to understand what the right security is for the product, and later-on test if it is met, one can conduct manual analysis or use VDOO's cloud-based platform that automatically creates a list of security requirements based on the specific device, its hardware, OS, and functional operations (for example, the supported connectivity protocol). Being aware of these security requirements before proceeding to the Design phase is highly recommended.

Phase 2 - Design

When designing the new product, among the questions that need to be answered are - what **Technologies** shall be used? What are the **Risks** in the design? What are the **Time and Budget** constraints?

In this phase, the system and software design documents are created, based on the requirement specifications defined in the previous phase.

The system design and the software design documents provide a detailed description of the various features and operations to support the functional requirements of the IoT product.

The design should clearly define all the architectural modules of the product, along with its communication and data flow representation as well as external and 3rd party modules.

When developing an IoT product, a **Chip Vendor** should be chosen as well as additional vendors for the peripheral components such as RAM memory, ROM memory, NAND flash, Audio, Switch, etc.

From a security perspective, the architecture and design should be reviewed using threat modeling techniques to point out identified potential threats, such as structural vulnerabilities, and to enumerate and prioritize them.

Phase 2 - Design

Threat modeling is based on four major steps:

Decomposing the application

Categorizing the threats

Ranking the threats

Mitigating the threats It is highly important to map out the security threats during the early design phase, as it can be very hard to mitigate them in later phases. For example, if we decide that the product will require a secure boot mechanism, in most cases we will need to support a Root-of-Trust (RoT) to implement this requirement. In such a case, support from the chip vendor is needed for the RoT integration. This integration will probably impact the basic software architecture design, therefore, early attention to this requirement is clearly more efficient.

To cover the security requirements in the design SDLC phase, VDOO's platform provides additional detailed information per each security requirement that includes severity level, the effort required for implementation, and how-to-implement guidance and reference to leading standards and regulations. This allows for better time planning.

Phase 3 - Development

Phase 3 - Development During the development phase, the code is built, tested, integrated and managed. The developer begins the coding according to what was defined in phases 1 and 2 above (Requirements and Design).

The first step in IoT product development is the **Hardware Bring Up**, that includes the CAD drawing of the proposed hardware layout, the editing of the electrical circuit cycles and the measuring of the power supply noise.

The second step is the **Software Bring Up**, that includes integration with the hardware team, setting up the initial board state by fuse burning, boot and kernel bring up, file system bring up (in case of Linux-based devices), and the development of the company's proprietary code.

The third step is the running of the **Static Analysis** tools on the entire codebase, a process that provides an understanding of the code structure and existing bugs, and helps ensure that the code adheres to industry standards.

The fourth step is the **Build Automation**, the process of creating the software build automatically. It has associated processes that includes fetching, patching, configuration, compilation, installation, and packaging.

The final steps in the development phase are the test code steps that include **Unit Tests**, **Black-Box Tests**, and **Hardware-In-The-Loop Tests**. This step includes a simulation of the sensors, actuators and mechanical components that connects all the I/O of the tested ECU, long before the final system is integrated.

Phase 3 - Development

From a security requirements perspective, it is important to integrate them into the CI process as early as possible, to ensure that the product is secure and also remains secure at each of the development cycles. An early integration of product security processes into CI dramatically reduces future overhead and delays related to after-the-fact security mitigations and late implementation. While most of the CI processes were designed with the software delivery speed as a top priority, integrating security too late may affect release dates significantly while initial effort for early integration of security could avoid these undesired delays.

Implementing critical security requirements at this point of the SDLC process (examples in the next paragraph) could be simplified by using the VDOO Analysis Platform. The platform automatically provides a device-specific report based on the firmware of the IoT product in question. The report includes, among other things, a full scope of security gaps (in different severity levels), a description of the resulting risks, reference to other standards and regulations, and step-by-step guidance for mitigation. In addition, the report includes 3rd party known software vulnerabilities as well as suspected zero-day vulnerabilities.

Phase 3 - Development

Below are examples of critical security requirements provided by the VDOO Analysis Platform, that should be implemented as soon as possible in the development phase:

Update operating system (kernel): Provide the customer with an up to date version of the operating system kernel. Older versions of the kernel may include critical security vulnerabilities. The decision on which kernel version to start the SDLC process with is critical, and in most cases irreversible. Our recommendation is to scan the first baked image with the VDOO Analysis Platform to be aware of security vulnerabilities relevant to the chosen kernel version at an early phase and learn of efficient ways to mitigate these risks.

Support secure updates: Secure update capabilities can be provided by vendor code, operating system package managers, or 3rd party products. Ensure that the update-service can update the kernel, the 3rd party software, and the in-house application components.

Add secure compilation flags: Enable security flags in compiler and linker settings, to protect against common exploits such as buffer overflows and heap corruption.

Phase 3 - Development

In compiler and linker settings, to protect against common exploits such as buffer overflows and heap corruption.

Ongoing use of the VDOO Analysis Platform as a plugin to an existing CI tool during the development cycles is recommended due to the following benefits:

Automation of the security processes: the VDOO Analysis Platform is integrated smoothly into the CI process with support for common CI automated build tools and automated allocation of security-related tasks.

Continuous feedback on the security level: with the VDOO Analysis Platform the development teams get an immediate notification with detailed explanations regarding the product security status and ways to fix the discovered gaps. This quick feedback is critical to keep on scheduled delivery timelines with a high security level.

Forcing a security check in all CI pipeline points: by integrating the VDOO Analysis Platform into the CI pipeline, security checks are forced in short development cycles. This leads to continuous protection from security risks even after minor code changes.

Phase 4 – System Testing

The focus of this phase is to verify that the product meets the technical, functional, and business requirements that were defined earlier, based on a series of tests that should cover the entire system.

A few of the major tests performed during this phase are the following:

Virtual Lab Tests to utilize virtualization services that simulate all of the hardware dependencies required to perform a full system test. Successful simulation of the real product, although not an easy task, will significantly save cost and time in later phases.

Acceptance Tests to evaluate the systems compliance with business requirements and assess whether it is acceptable for delivery.

Once the tests above are completed, **Physical Lab Tests** are performed to allow a complete system test of the real product in its real environment.

The last test is the **Certification Test** where certification-related tests are performed, such as safety, security, and network protocols.

Phase 4 – System Testing

From the security perspective, during the system testing phase, there are a few security guidelines that should be met:

- Identify critical security problems by running static and dynamic analysis of the complete final firmware image.
- Run penetration tests to check how the product handles various abuse cases such as malformed input handling, authentication/authorization bypass attempts, and overall security posture.
- Ensure that security assumptions specified during the design phase are still relevant.

Phase 4 – System Testing

The VDOO Analysis Platform helps to meet the aforementioned security requirements and provides a valuable security gap analysis report at this point of the SDLC process, as the integrated firmware image is being analyzed entirely. It is clear, yet important to mention that as the release date gets closer, non-critical implementation changes will not be applied to the product. The VDOO Analysis Platform classifies the requirements automatically in a way that the user can focus only on the security requirements that are critical and safe enough for implementation given the relatively close release date. This will usually result in requirements that are related to critical mitigations and hardening and not architectural or design changes. For example:

Ensure that source files and headers are removed from the device: Source files and headers are much easier to analyze than compiled binaries and help the attacker identify weaknesses in the product.

Remove development features from firmware: If a device uses production tests, calibration software, and other development and debugging features used during manufacture, erase, remove or block this functionality before the product is shipped. This protects against malicious use of such features in the field by attackers with physical or network access.

Phase 5 – Development and Maintenance

The main focus of this phase is to make the system operational in a live environment and to maintain system operational stability.

The first step in the Deployment and Management phase is the **Manufacturing** step that in most cases is performed by 3rd party manufacturing companies.

The second step is the **Production Validation** where the device is validated under a production environment.

The third step is the **Release** step where the product is delivered to the market.

The fourth and last step that closes the entire SDLC process is the **Steady State** step, where the product is being used by customers and being maintained and monitored for its performance and for the user experience.

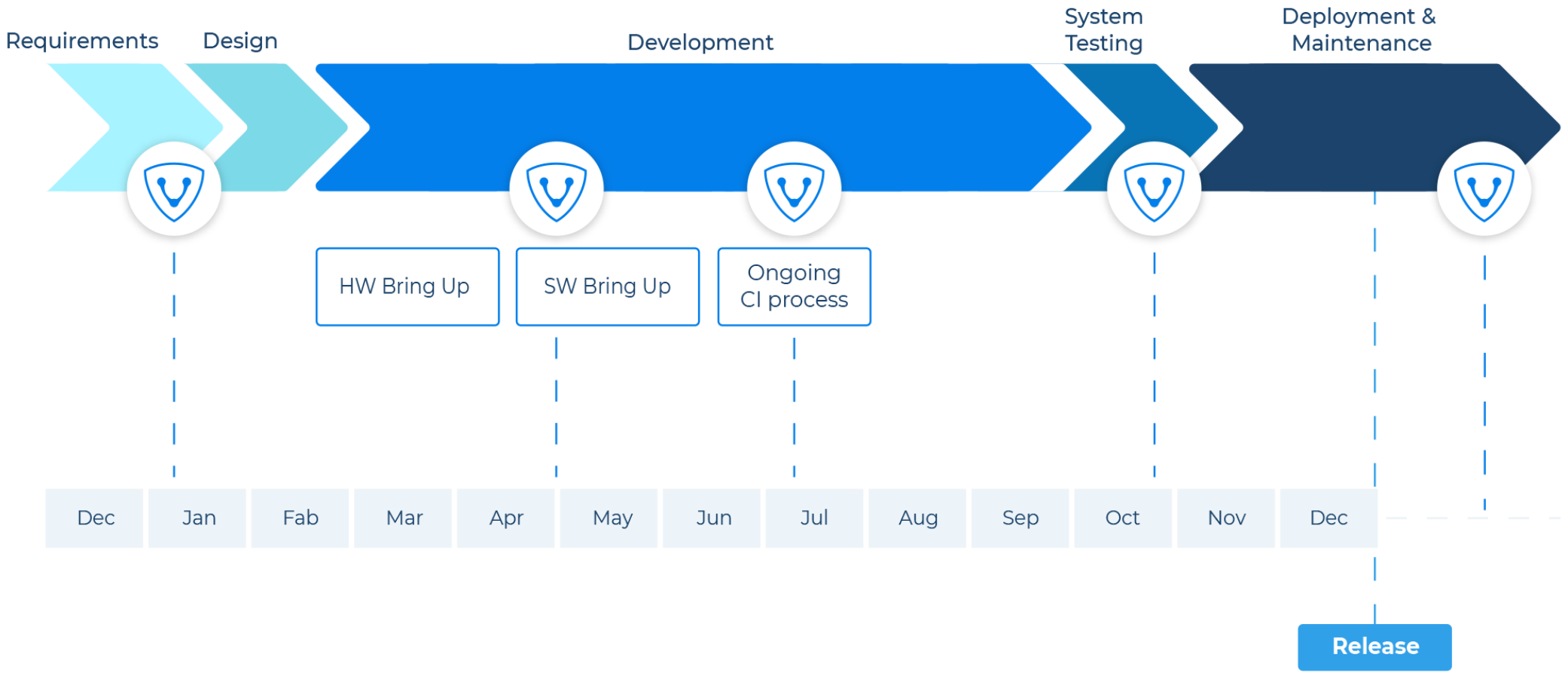
From a security requirements perspective, it is important to verify that the product manufacturing follows security standards guidelines (as defined usually by industry) and that the product is shipped with no unwanted physical ports open. Finally, it is required to keep the product software updated to ensure a healthy security system, with special attention to 3rd party vulnerability management.

Phase 5 – Development and Maintenance

Implementation of these security requirements can be performed manually before and after the device release, or could be conducted automatically using the VDOO Analysis Platform on the product image. VDOO's database is continuously updated for new threats and vulnerabilities as well as common SW components and libraries. Therefore, it is highly recommended to analyze the firmware also in the post-deployment of the product, to check for up-to-date device-specific vulnerabilities (including 3rd party) based on the product hardware and software, to enhance the vulnerabilities management.

The following scheme describes the typical SDLC process for an IoT product, with the recommended points where using the VDOO platform optimizes the security integration process. The first use of the VDOO platform is during the Requirements and Design phase to provide the necessary device-specific security requirements list. The second use is during the early Development phase of Software Bring Up, and again during the ongoing CI process. The fourth use is during the System Testing phase where certification tests are required. The fifth use is during the Deployment & Maintenance phase in the Steady-State step, where additional security analysis is required to ensure that the device's security level is maintained even after its release.

Phase 5 – Development and Maintenance



Conclusion

The integration of security into the SDLC process in each and every phase is essential for the development of secure software while reducing the total cost and effort involved in security implementation. Continuous security integration into the SDLC process as described in this article will continuously improve the product security and will also increase the probability that the product meets business goals.

Contact us to learn more about how the VDOO platform can help in integrating security into an IoT SDLC.

Maor Vermucht, VDOO