LABORATOIRE DE CONCEPTION ET D'INTEGRATION DES SYSTEMES

Expression of Needs and Identification of Safety Objectives for a connected medical pump

EBIOS

BRESCH Cyril

23/11/2018

50 Rue Barthélémy de Laffemas, 26000 Valence. Tel 04 75 75 94 00

Modification History

Date	Modification Object	Status
23/11/2018	Creation of the Document	Validated

Table of Contents

Introduction

The Very Secure Pump Model is an open platform modelling a simple infusion pump using Bluetooth Low Energy technology. The Very Secure Pump Model has been designed within the Serene IoT project which aims at contributing to develop high quality connected care services and diagnosis tools based on Advanced Smart Health-Care IoT Devices. This platform intends to provide a framework for security evaluation, tailored for countermeasures development against the numerous security flaws related to medical devices. These vulnerabilities have been identified in various papers in the literature that will be referred through this report. The vulnerabilities assessments established here constitute a background for the research lead on the security of life critical systems: it should provide insights on the threats such systems are exposed to and their likelihood, and provide an orientation to the security work to ensure addressing the identified flaws. The outline of this document is based on EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) the methodology defined by the National Cybersecurity Agency of France (ANSSI, Agence Nationale de la Sécurité des Systèmes d'Information [1]), with slight adaptations to the specific context concerned here. Nevertheless, it should follow the main guidelines of most international vulnerabilities analysis report writing convention.

The report is divided in two parts, as described in EBIOS [1]:

- **The Medical Pump Model Presentation**, describing the VSMP platform, its functionality, the scope of this analysis.
- **Risk Analysis**, bringing together the five evaluations recommended in the eBIOS method:
    - **Analysis of the Context**: Manage risks over the long term and develop a policy.
    - **Analysis of Feared Events:** Identifying the threats faced by the Medical system and their impact (regardless of how they do happen) and highlighting the most critical ones.
    - **Threats Scenarios:** Depicting the attack scenarios that can lead to the aforementioned threats, with an estimation of their cost, difficulty and implementation time.
    - **Risk Analysis:** Estimating the impact and the likelihood of the aforementioned scenarios given the security implemented in the system.
    - **Security Measures:** Proposing security solutions for the unneglectable risks identified in the previous section.

The Very Secure Pomp Model platform aims to be representative of a classic infusion pump system, which implies that the vulnerabilities assessment established in this report apply for any system implemented in a related context. Some of the vulnerabilities identified are specific to either Bluetooth Low Energy and may not apply to other medical devices. This document provides good insights on the general security of life critical systems, and help security developers and researchers to quickly identify the most critical flaws to address on such systems.

I. The Medical Pump Model Specifications

1. Global Presentation of the Medical Pump Model

The Very Secure Pump Model (VSPM) is an open secure medical pump demonstrator which development started in September 2017, in LCIS laboratory (Grenoble Institute of Engineering, Valence, France). The objectives of this platform are threefold:

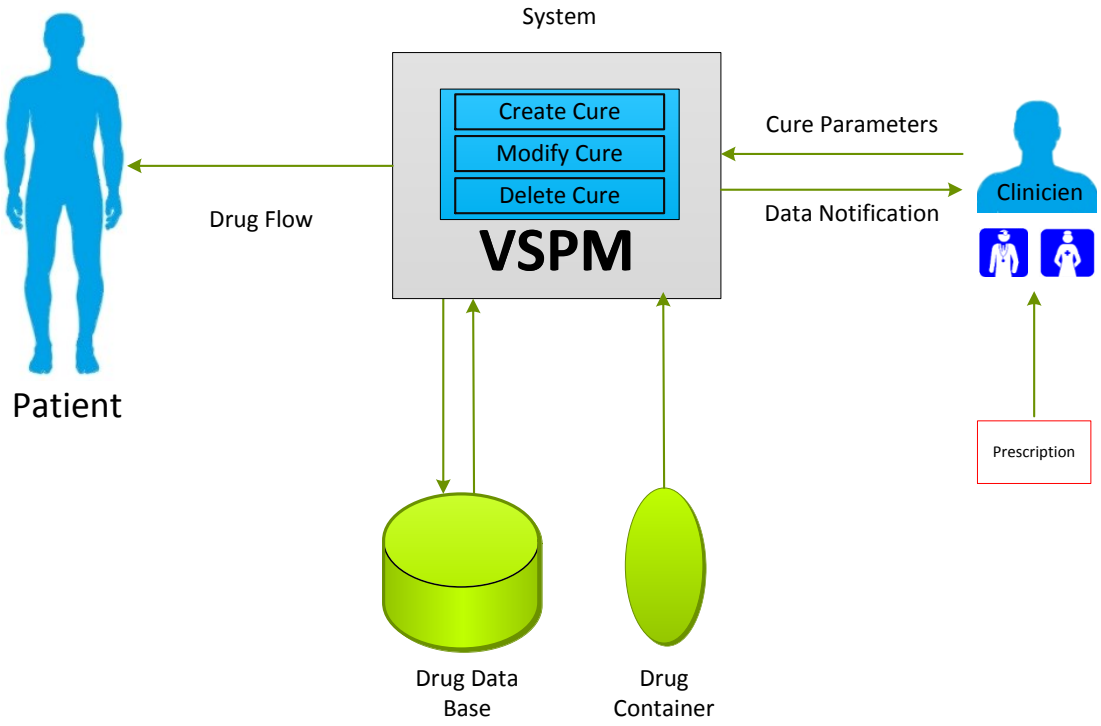
- Establish and provide a set of documents of good security/safety practices for the development of a life critical embedded application. Our software architecture document is also open source and provides all the details related to the medical pump.

- Provide an abstract representation of the software used in a typical infusion Pump. This open source base can be extended and improved for specific implementation/demonstration.

- Provide an open-source medical platform that can be used to demonstrate and expose the exploitation of hardware and software vulnerabilities. For example, at the hardware level, our technology uses encryption algorithms such as AES. It is therefore possible to carry out side channel attacks to leak cryptographic keys. At the software level, a set of software layers and protocols allow the pump to operate and communicate using Bluetooth Low Energy. Thus, these network interfaces can be used to demonstrate the efficiency and the power of a remote buffer overflow attack. Finally, this highly modular platform also aims to be used to evaluate and validate security countermeasures in critical physical systems.

As defined above, the Very Secure Pump Model (VSPM) is an open source cyber physical system that models the dosage and performance of a drug administered to a patient in real time. The VSPM currently allows only one operating mode to be simulated: "*constant mode*". In this operating mode the pump injects a volume of medication during a constant period of time. The injection rate of the drug is therefore directly proportional to the volume and time set at the start of a cure. Once the cure is launched, the injection of the drug is performed at a constant rate. Once the injection is completed, the pump stops by itself. It is therefore possible to program a new treatment. Throughout the duration of the care, the clinician has the possibility of:

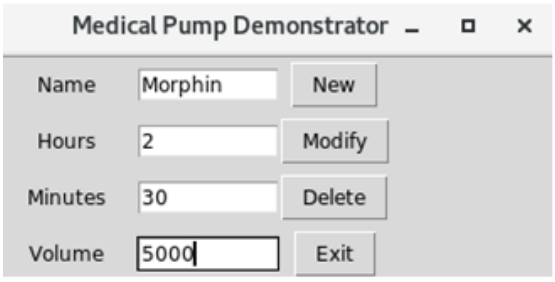
- **DELETE** the cure.
- **MODIFY** the time data to extend or reduce care.
- **MODIFY** the volume data to increase or decrease the flow rate.

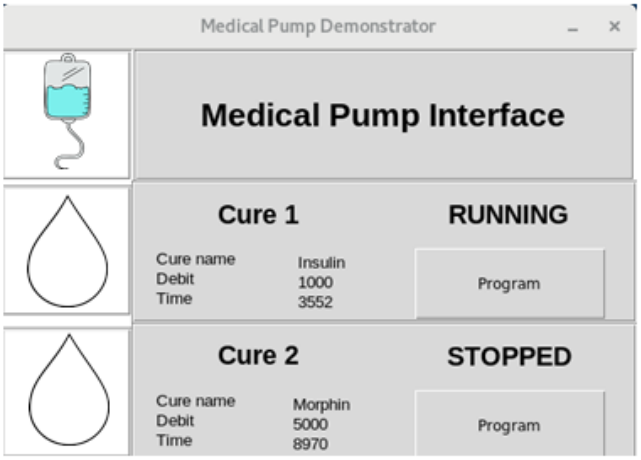


In the system above, the clinician unlocks the programming interface of the pump in order to provide the patient with care. To do this, it fills in all the information concerning the drug to be injected into the pump interface, connects the device with the patient, loads the drug into the compartment provided for this purpose and finally starts the treatment with the prescribed information. During the entire duration of the treatment, it is possible for the clinician to monitor the activity of the pump on the interface. In addition, all pump events are logged and can be viewed in real time.

2. Services Provided by the Abstract Pump

The VSPM is a simple abstract model of a connected pump. Its purpose is to be simple to install and simple to display. This model consists of embedded software on a microcontroller as well as a cross-platform software interface (Linux, Microsoft) allowing communication with the connected device. Once connected to a power supply, the pump starts and waits for a BLE communication. This allows the clinician to start the interface that will initiate the communication with the pump. The graphical user interface then allows the pump to be programmed using the BLE. A representation of the interface is given below.





Thanks to the interface, the clinician can initiate a treatment. To do this, the clinician clicks on the “Program” button. A window will then open proposing to enter the name of the drug to be injected, the volume to be injected and the duration of the simulated treatment. The interface then builds the BLE packets and sends all the information to the pump in order to program it.

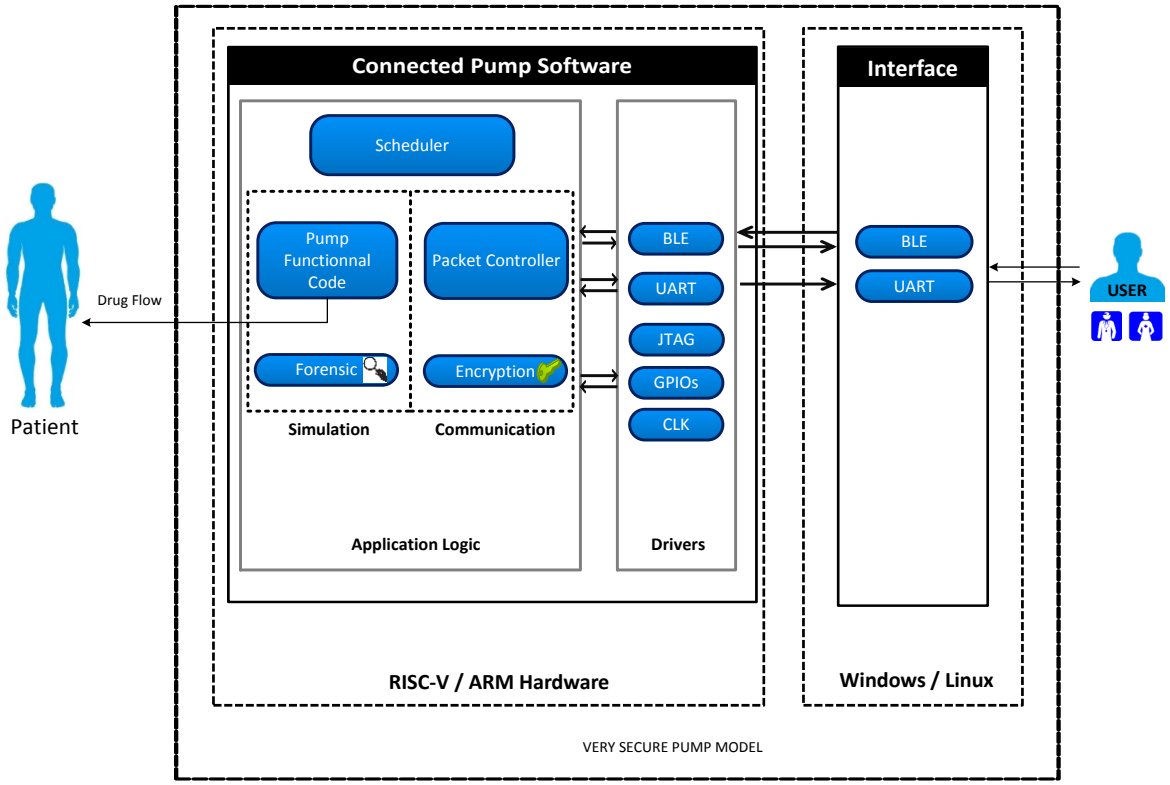
Once the pump has been carefully programmed, the injection rate of the drug is computed independently. Several pieces of information are sent back in real time to the interface via the BLE communication. Indeed, the pump returns the time remaining before the end of the treatment, the volume remaining in the drug bag, as well as the name of the drug in progress. This data, which can be consulted in real time, makes it possible to detect any dysfunctions during the cure. When the treatment time has expired, the embedded software stops the injection independently. It is then possible to restart a treatment with new parameters.

During the simulation of the treatment it is possible to modify the injection data. This operation is carried out via the interface. The clinician must click on the "Modify Cure" button and enter the new treatment values. Once again the interface builds the correct BLE packets to modify the current cure and sends all the information to the pump in order to reprogram it. Obviously, if no cure is simulated it is not possible to use/tamper the “modification” button. Similarly, for safety reasons it is not possible to change the name of the drug during treatment. In order to change the name of the current treatment, it must be deleted first and recreated with a new name.

Finally, the interface also allows the clinician to delete a current treatment and resets the status of the pump. For that, the clinician must click on the "Delete Cure" button and enter the new name of the cure he wants to delete. Then the interface builds the correct BLE packets to stop the current cure and reset the state of the pump. Obviously, if no cure is simulated it is not possible to use/tamper the “delete” button.

3. Structure of the Medical Pump System

This section describes the software and hardware structure of the VSPM. The figure below is a global representation of the pump structure. The following sub-sections detail the operation and orchestration of the modules with each other. The main interface is supported on both Windows and Linux operating system. The embedded system software is using Hardware Abstraction Layer for drivers and has been designed to run on a STM32 Nucleo board.

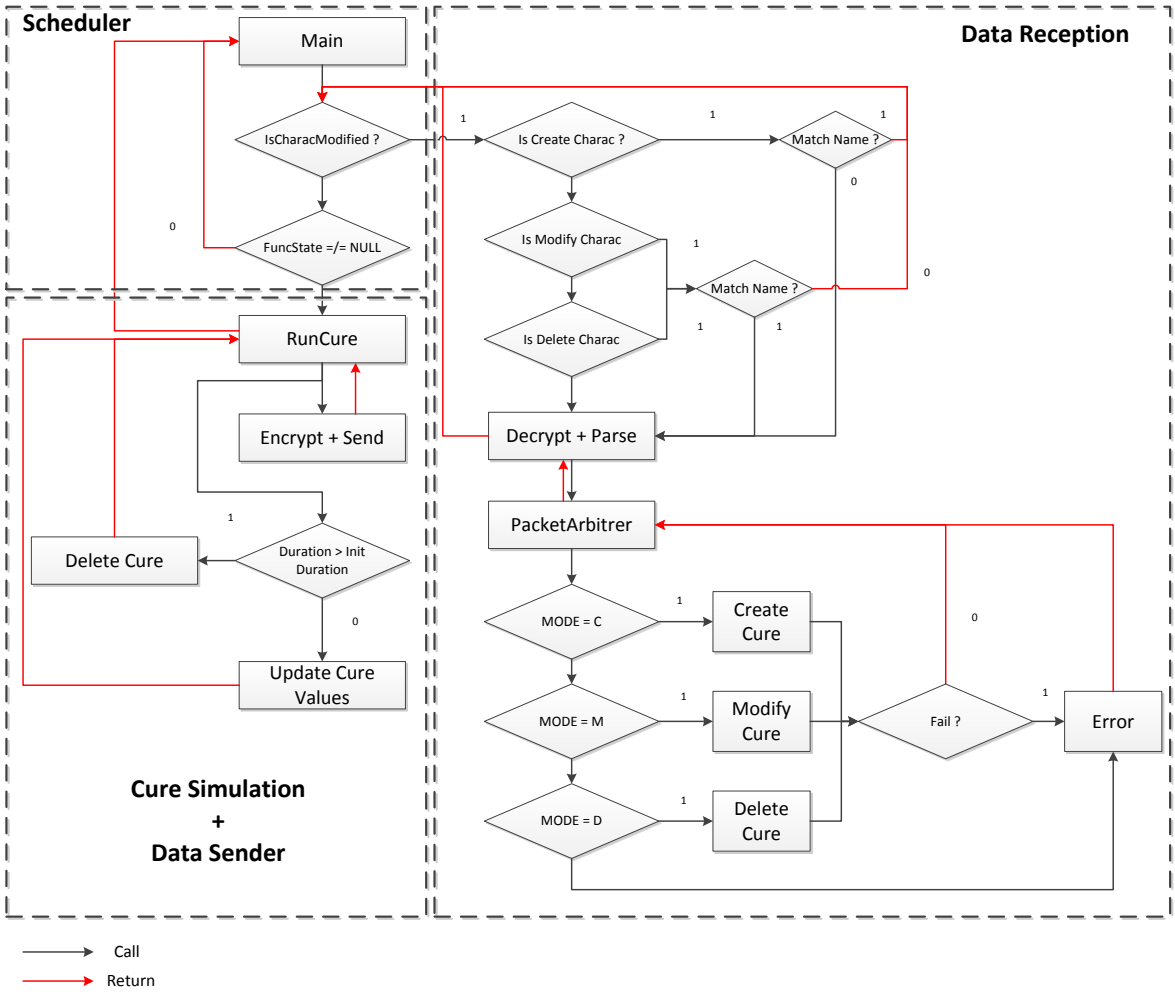


The user interface is a software component running on a host computer (Windows or Linux). This interface uses the UART and Bluetooth Low Energy 4.2 protocols to communicate with the embedded system. The Bluetooth Low energy protocol is used to send/receive data to the embedded system while the UART protocol is used for debugging purposes. On the pump side, the reception of the packets from the interface is managed by the drivers of the HAL library coupled with the packet manager. Depending on the action sent in the packet, the embedded software simulates the operation of the connected pump. During the simulation the packet manager is called to send back the information of the current treatment. Indeed, all real time values are sent back to the interface through Bluetooth Low Energy communication. Finally all actions are logged in the pump and can also be observed using the UART interface.

a. The Application Logic

The entire embedded application has been entirely developed in C language. VSPM also uses third-party drivers provided by the HAL library. As VSPM models the behavior of a critical embedded system, it has been modeled to follow the best practices in term of memory usage. Indeed, dynamic allocation is widely considered taboo in safety-critical embedded software. Dynamic memory allocation can introduce dangerous memory leaks, memory fragmentation, unpredictable performances and memory overhead. According to the DO-178B standard the use of dynamic memory allocation is strictly forbidden in safety critical embedded avionics code. The VSPM aims at respecting this standard and provide a static and safe memory application with no unpredictable memory behavior. The control flow diagram of the application is presented below. It is based on a state machine always returning to its initial state: main. Thus any memory allocated by a state is deallocated when returning to the initial state. There is no recursive function call or nested state.

- Below is an overview of the control graph of the Embedded Software.

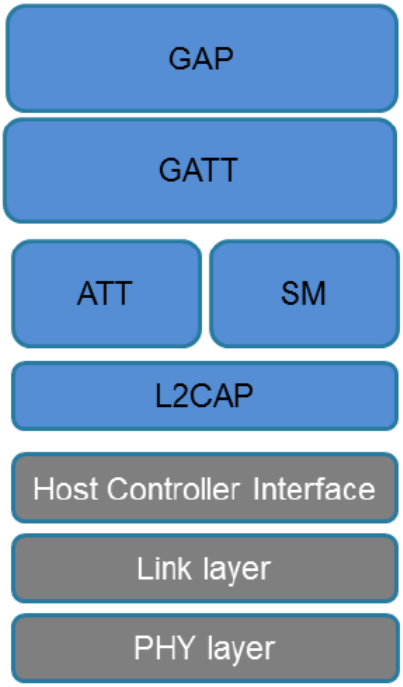


- The Scheduling part is in charge of checking if some Bluetooth Low Energy characteristics have been updated and launch a simulation if a cure is already configured.

- The Data reception part is called by the scheduler when a Bluetooth Low Energy packet updates a Bluetooth Low Energy characteristic. More details about the Bluetooth Low Energy protocol is provided in the following dedicated section. All the data in the packet are identified by the Parser and the according action is decided by the packet arbiter. A static state pointer is modified according to the programming packet. This state pointer is then called by the scheduler after the data reception process.
- Simulation: At the initialization of the pump the state pointer is null because no cure is currently running on the pump. If the state pointer is non-null it means that the pump has been already programmed. Hence, the simulation is starting sending back the current state of the pump (resting time before the end of the treatment, resting volume, name of the cure).

b. The Drivers: Bluetooth Low Energy Protocol

The Very Secure Pump Model is currently using the Bluetooth Low Energy 4.2 Stack displayed below. This wireless protocol has been chosen for the VSPM due to its low power and low cost, low bandwidth and low complexity features. Furthermore, due to the IoT trend, more and more semiconductor manufacturers are implementing BLE in their chip to make them used in real-world applications tight on energy with minimal budget.



In the above stack, the GATT layer provides several services used by the connected pump. Each service is defined with their given UUID. For the VSPM a first service called “command” is dedicated to the commands that are sent by the interface to the pump. The second “check” service is used by the pump to send back real time data to the interface. A service discovered with its UUID may handle some characteristics. Characteristics are defined attribute types that contain a single logical value. In the “command” service, three characteristics are defined: **Create Cure**, **Modify Cure** and **Delete Cure**. These previous characteristics can be written and comply with the specifications in

the previous sections. Likewise, the "Check" service provides three characteristics that can be read: **Name**, **Volume** and **Duration.** These characteristics are used to send back real time values to the interface.

Service/Characteristic	Properties	Length (bytes)	Type	Unity
Command Service				
Create Characterictic	Write	48	String	
Modify Characterictic	Write	48	String	
Delete Characterictic	Write	48	String	
Check Service				
Name Characterictic	Read, Notify	32	String	
Volume Characterictic	Read, Notify	4	String	ml
Duration Characterictic	Read, Notify	8	String	s

Service/Characteristic	UUID
Command Service	fcb43012-cbc4-11e8-a8d5-f2801f1b9fd1
Create Characterictic	fcb432ec-cbc4-11e8-a8d5-f2801f1b9fd1
Modify Characterictic	fcb4344a-cbc4-11e8-a8d5-f2801f1b9fd1
Delete Characterictic	fcb4358a-cbc4-11e8-a8d5-f2801f1b9fd1
Check Service	fcb43abc-cbc4-11e8-a8d5-f2801f1b9fd1
Name Characterictic	fcb43bde-cbc4-11e8-a8d5-f2801f1b9fd1
Volume Characterictic	fcb43d0a-cbc4-11e8-a8d5-f2801f1b9fd1
Duration Characterictic	fcb4417e-cbc4-11e8-a8d5-f2801f1b9fd1

Key management & Encryption

The Very Secure Pump Model is sending a considerable amount of private data through the BLE channels. To prevent potential eavesdroppers including malicious hackers, trackers or communication providers, the embedded system is using end to end encryption. End-to-end encryption is intended to prevent data being read or secretly modified, other than by the true sender and recipient(s). Our model is mainly using AES 128. A first version of the system is using the AES 128 algorithm provided in the BLE 4.2 Stack. A second version of the pump is using a Tiny 128 AES validated by the NIST. In both cases the key is assumed to be put in a secure element of the STM32 board. As we consider that the secure communication is using a symmetric algorithm. The symmetric key must only be known by the provider of the embedded system.

For the Bluetooth Low Energy the secure communication is established by the generic access profile and the secure manager in the Bluetooth Low Energy stack. The Bluetooth low energy link layer supports encryption and authentication by using the counter mode with the CBC-MAC (cipher block

chaining-message authentication code) algorithm and a 128-bit AES block cipher (AESCCM). When encryption and authentication are used in a connection, a 4-byte message integrity check (MIC) is appended to the payload of the data channel PDU. More details about the key exchange and the implementation of the secure connection pairing model using elliptic curves can be found in the documentation BlueNRG-1, BlueNRG-2 BLE stack v2.x programming guidelines of STMicroelectronics.

4. Software and Hardware Stack

II. Security Analysis

1. Context Analysis

Context:

Given the specifications presented in the previous sections, this report will examine the vulnerabilities of the Very Secure Pump Model at three different levels:

- The vulnerabilities related to the implementation of the VSPM at the hardware layer
- The vulnerabilities related to the implementation of the VSPM at the software layer
- The vulnerabilities related to the use of BLE 4.2 communication protocol.

This security analysis mainly focuses on the backend of the entire model (the embedded system). All the attacks that may take place on the front end like the interface are considered out of scope of the study. Thus, we assume that the interface is running in a safe operating system environment and its integrity has not been tampered.

The VSPM is not meant for a particular industrial application, and is rather a general representation of life critical embedded systems in this study. Estimating the impact of threats and their severity is a huge part of vulnerability analysis, and defining metrics is required to do so. However, providing metrics for the severity or impact of threats has a limited relevance without additional hypothesis. Thus, this section intends to give a brief background on the most common use case of this life critical device.

From an applicative point of view, the most important characteristic of the VSPM is to cure patient remotely.

Some examples are:

- **Home Caring:** Enable patient care from home while sending back data to a cloud from a local programmer. This option therefore allows the doctor to monitor the patient's treatment remotely.
- **Hospital Caring:** In this case the patient is cured in a hospital. Real-time data are therefore reported locally. Patient follow-up can therefore be done from a local interface improving the availability of medical staff.

Given these properties, the Very Secure Pump Model has the following characteristics:

- High accuracy requirements
- Real Time requirements
- Availability requirements
- Safety and Security requirements

Regarding security, it can be observed that several properties can improve the trust of this life critical device:

- **Integrity:** Guarantees that the embedded system software can be trusted and has not been maliciously tampered.
- **Authenticity:** Guarantees the identity of the VSPM communicating with the remote interface and vice versa. This feature prevents malicious pump or gateway to spoof the identity of the VSPM or the remote entity.
- **Availability:** The system must be accessible from the moment it is used by the patient
- **Confidentiality:** All assets related to the intellectual property of the system must be protected.
- **Privacy:** Each piece of data regarding the privacy of the user is ciphered using cryptographic algorithm. Thus, in case of robbery or eavesdropping it is not possible to retrieve the data without the encryption key.
- **Safety:** The system needs to be safe and validated by a certified authority. In addition the system needs to be robust. Indeed, in case of failure or bugs the application has to recover without putting the patient's life at risk.

We made the following hypothesis for the further analysis:

- The system cannot be run safely if the integrity of the embedded application has been damaged.
- The system cannot execute commands/send data from/to an untrusted remote entity.
- Rogue command execution can lead to damages and/or human harm.
- In case of damages, the life of the patient can be harmed.
- In case of non-accessibility during use, the system may endanger the patient's life.
- In case of damages, the loss caused by the damages is most likely higher than the price of the whole critical system.
- The VSPM is put in an untrusted area.
- Privacy is not the main concern for safety but remains required. The system cannot communicate without encrypting data.

Metrics:

Given the previous hypothesis, the metrics that are used in this analysis are defined as following:

Security Scale:

Criteria	Definition
Integrity	Property of accuracy and completeness of assets
Availability	Property of accessibility when assets is needed
Authenticity	Property of assuring to ensure that information comes from the source it claims to be
Confidentiality	Property of assets to be accessible only to authorized users

Severity Scale:

Level	Qualification	Definition
1.	Insignificant	The system will overcome the

		impacts without difficulties
2.	Limited	The system will overcome the impacts despite some difficulties
3.	Significant	The system will overcome the impacts with serious difficulties
4	Critical	The system will not overcome the impacts

User Severity Scale:

Level	Qualification	Definition
1.	Insignificant	The patient is not affected
2.	Limited	The patient is affected but its life is not in danger
3.	Significant	The patient is affected but the danger to his life is not imminent
4	Critical	The patient is affected and the danger to his life is imminent

Likelihood:

Level	Qualification	Definition
1.	Minimal	Very unlikely to happen
2.	Significant	Can happen (again)
3.	Highly Significant	Can happen (again) one day
4.	Maximal	Going to Happen soon

These metrics are relevant as long as the hypotheses are verified; the severity scale should be adapted for a system following a very different applicative scheme. Nevertheless, the analysis regardless of the metrics used will remain relevant.

Assets Identification:

Assets are information, capability, features, financial or technical resources that may be damaged, lost or disrupted. The identification of assets also involves the study of the processes computed by the system.

Process	Assets Concerned	Actor
Pump Booting and Initialization	• Electronic circuit of the device • Battery • Bootloader • Embedded Application • Software Version • Electronic Records	Clinician via Reset Button
Initiation of Communication	• Pin Code • Cryptographic Keys • Certificates • Device UUID • Electronic Records	Clinician via Interface

Creating Cure	• Network Packets • System Logic • Cryptographic Keys • Cure Volume data • Cure Duration data • Cure Name data • Electronic Records • Patient's Prescription	Clinician via Interface
Modifying Cure	• Network Packets • System Logic • Cryptographic Keys • Cure Volume data • Cure Duration data • Cure Name data • Patient's Prescription • Electronic Records	Clinician via Interface
Cure Processing	• Cure Name Data • Cure Volume Data • Cure Duration Data • System Logic • Patient's Data • Electronic Records	Very Secure Pump Model
Stopping Cure	• Network Packets • System Logic • Cryptographic Keys • Cure Volume data • Cure Duration data • Cure Name data • Electronic Records	Clinician via Interface

Thread Sources:

Source	Considered	Examples
User of the Pump	No	
Internal attacker with limited skills	No	
Internal attacker with significant skills	No	
Internal attacker with unlimited skills	No	
External attacker with limited skills	Yes	Script kiddies, Skilled Staff
External attacker with significant skills	Yes	Cyber-Criminals, Terrorists
External attacker with unlimited skills	Yes	Governments
Internal person, limited skills, no desire to harm	Yes	Clinician
Internal person, significant skills, no desire to harm	Yes	Clinician, Administrator,
Internal person, unlimited skills, no desire to harm	No	
External person, limited skills, no desire to harm	Yes	Visitor
External person, significant skills, no desire to harm	Yes	Visitor, Internet Provider, Hosting Company, Network Manager
External person, unlimited skills, no desire to harm	No	

Non-targeted viruses	Yes	Random Virus
Natural disaster	Yes	Natural phenomenon (lightning, wear and tear...)
Animal activity	No	
Internal event	Yes	Power Failure

2. Feared Events

Feared Scenario	Threat Source	Impacts	Patient's Impact Severity
The electronical circuit is unavailable			
The electronical circuit is altered			
The electronical circuit is compromised			
The embedded software is altered			
The embedded software is compromised			
The software version is altered			
The electronics records are unavailable			
The electronics records are altered			
The electronics records are compromised			
The certificates are stolen			
The certificates are altered			
The certificates are compromised			
The Cryptographic Keys are stolen			
The Cryptographic Keys are altered			
The Cryptographic Keys are compromised			
The system logic is unavailable			
The system logic is altered			
The system logic is compromised			
Duration, Name,			

Volume internals data are unavailable			
Duration, Name, Volume internals data are altered			
Duration, Name, Volume internals data are compromised			
The network packets are unavailable			
The network packets are altered			
The network packets are compromised			
Patient's Prescription is stolen			

3. Threat scenario analysis

Threat scenario	Threat source	Likelihood	Security Impact	Severity	Patient's impact
Malicious hardware injected in the motherboard					
Extraction of the proprietary code					
Malicious code injected in the software					
Exploitation of a bad implemented cryptographic algorithm					
Rogue command execution					
Patient's information hijacking					
Communication spoofing					
Pump Rogue Access Point					
Denial of service					
Software Bug exploitation					

4. Risks Analysis

5. Security Measures