Eindhoven University of Technology

MASTER

# On security of implantable medical devices

Slobbe, J.

*Award date:*
2013

Link to publication

**Disclaimer**
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

# On security of Implantable Medical Devices.

**Master thesis for obtaining the master of science degree at the TU/e**

J. Slobbe*

March 14, 2013

Graduation committee:
Prof. dr. S. Etalle
S. Hernando
dr. B. de Weger

Technische Universiteit Eindhoven
Den Dolech
5612 AZ Eindhoven
Tel: +31 40 247 91 11

Deloitte
Laan van Kronenburg 2
1183 AS Amstelveen
Tel: +31 88 288 28 88

Jeroen Slobbe
Hoge Rijndijk 316
2342 AN Leiden
Tel: 06 49 717 934

*j.slobbe@student.tue.nl

This paper is the final milestone needed to graduate at the Technical University of Eindhoven for the Information Security Technology program. This thesis is supervised by Prof. dr. Sandro Etalle from the TU/e and Sergio Hernando from Deloitte Security & Privacy.

## Acknowledgements

I would like to take this opportunity to thank several people for their advice and support. First of all Sandro Etalle for supervising and advising me during the master project. His help greatly helped me improving my thesis in every possible aspect. Secondly I would like to thank my supervisor from Deloitte: Sergo Hernando, for his advice, knowledge and the interesting discussions we had during the process of writing my thesis. Thirdly I would like to thank Ir. Stas Verberkt, Ir. Bert Tilmans and Daan Muller for reviewing my thesis and providing me with very useful comments. Fourthly I would like to thank my colleagues from Deloitte for discussions, feedback and the good time we had during my internship. I would also like to thank all interview participants, since I promised to keep them anonymous I cannot name them specificity here but I am very grateful for the experiences, knowledge and honest answers they shared with me! Finally I would like to thank my parents, brother and friends for supporting me during the process of writing my thesis.

**Abstract**

Millions of European citizens depend on Implantable Medical Devices to stay alive or to improve the quality of their life. Although implantable medical devices provide a solution for various disorders they also introduce the threat of an unauthorized attacker harming a patient via his or her implantable medical device. To make the problem even more complex, implantable medical devices need to be accessible by any healthcare professional in case of emergency and implantable medical devices are limited in their resources. For example implantable medical devices have limited cryptographic capabilities, limited storage space and limited processing power. Taking in account these limitations, we investigate: how to improve implantable medical device security.

To answer this question we start by discussing several implantable medical devices. Based on the architecture of a pacemaker, a generic insulin infusion pump and a deep brain stimulator we provide an abstract implantable medical device architecture. For this abstract architecture we enumerate desirable security properties and implantable medical device characteristics.

Secondly we discuses 10 vulnerability types: Weak or non-existing authentication, Limited battery capacity, Wired communication, Unencrypted communication, Weak encryption, Software / firmware vulnerabilities, Electromagnetic interference, Traffic analysis, Social engineering and unsecured physical access. For each vulnerability type, we explain how an attacker could use it to exploit an implantable medical device and what the impact of successful exploitation might be. We also discuss the type of attackers who would be capable of performing the described attacks.

We discuss three open security standards to investigate to what extent they are applicable to implantable medical device security. We discussing the applicability of the OWASP verification standards, the 20 CSC v4.0 from SANS and the relevant parts from ISO/IEC 27001:2005.

Then we perform a security assessment on the cardiac resynchronization therapy device. We analyse the manual, discuss methods to connect to the device and try to find methods to hack into the device. We contribute to implantable medical device security research by demonstrating that device vulnerabilities can be triggered by the exploitation of healthcare professional equipment. Based on the findings and the desired security properties we give an opinion about the security status of the cardiac resynchronization therapy device.

Finally we contribute to implantable medical device security by proposing an implantable medical device security assessment methodology.

**Keywords:** Implantable Medical Devices, pacemaker, insulin pump, security assessment, medical device attacks, medical device vulnerabilities, risk assessment.

# Contents

# List of Figures

# List of Tables

## Acronyms

**ADC** Analog to Digital Converter

**AED** Automated External Defibrillator

**APT** Advanced Persistent Threat

**ASVS** Application Security Verification Standard

**CIA** Confidentiality, Integrity, Availability

**CRT** Cardiac Resynchronization Therapy

**DAC** Digital to Analog Converter

**ECG** Electrocardiogram

**EMI** Electromagnetic interference

**FPGA** Field-programmable gate array

**ICD** Implantable Cardiac Defibrillators

**IED** Intelligent Electronic Device

**ISO** International Organization for Standardization

**IMD** Implantable Medical Device

**MASL** Medium Attacker Skill Level

**MICS** Medical Implant Communication Service

**MTU** Master Terminal Unit

**NCSC** National Cyber Security Center

**OSI** Open System Interconnection

**OWASP** Open Web Application Security Project

**PHI** Personal Health Information

**PKI** Public Key Infrastructure

**PLC** Programmable Logic Device

**RF** Radio Frequency

**SCADA** Supervisory Control And Data Acquisition

**SDR** Software Defined Radio

**TU/e** Technische Universiteit Eindhoven

**UHD** USRP hardware driver

**UDI** Unique Device Identification

**USRP** Universal Software Radio Peripheral

# 1. Introduction

Millions of European citizens depend on Implantable Medical Devices (IMDs) to stay alive [28] or to improve the quality of their life. Patients use insulin pumps for the treatment of diabetes, deep brain implants for the treatment of Parkinson and pacemakers for treating rhythm disorders. Although IMDs provide a solution for various disorders, IMDs also introduce the threat of an unauthorized hacker harming a patient via his or her IMD [62, 38, 42, 37, 23]. In this thesis we investigate how we can improve IMD security. When we told people about our research topic, people asked: Why are you researching this topic? Who is ever going to abuse security vulnerabilities in an IMD? In response, we gave two cases of people acting maliciously without a particular aim. In 1982 there was the Tylenol crisis[1]. An unknown suspect put 65 milligrams of deadly cyanide into Tylenol capsules. Seven people died because of that and market share from the Tylenol capsules company went from 37% and a revenue of 1.2 trillion a year, to a small 7% share [94]. In 2008 a group of malicious internet users posted a flashing javascript code on the internet forum of the Epilepsy foundation[2]. Members of the forum reported physical damage because of the action. We could learn from these examples that there are people with a malicious intent and those malicious people may eventually target medical equipment [37]. Hopefully, the security community improves IMD security and mitigates security risks before disaster, as described above happens!

## Introduction to IMDs

An IMD or *active IMD* is defined by European Counsel Directive 90/385/EEC [25] as: "an active medical device which is intended to be totally or partially introduced, surgically or medically, into the human body or by medical intervention into a natural orifice, and which is intended to remain after the procedure" A medical device is defined in the same directive [25] as: "any instrument, apparatus, appliance, software, material or other article, whether used alone or in combination, together with any accessories, including the software intended by its manufacturer to be used specifically for diagnostic and/or therapeutic purposes and necessary for its proper application, intended by the manufacturer to be used for human beings for the purpose of:

1. diagnosis, prevention, monitoring, treatment or alleviation of disease,
2. diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap,
3. investigation, replacement or modification of the anatomy or of a physiological process,
4. control of conception,

and which does not achieve its principal intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means;". IMDs are able to transmit data and can be reprogrammed to adjust (therapy) settings [37]. To be able to perform these operations, IMDs make use of telemetry. In 1987 NASA [66] defined telemetry as: "a technology that allows data measurements to be made at a distance". IMDs use short range telemetry to communicate wirelessly from inside the human body to external equipment. It is necessary to communicate wirelessly because it is undesirable to perform surgery for data retrieval or therapy modification [43]. To facilitate this connection and to prevent interference from other non medical usage of this band, the European Counsel [19] reserved the radio frequency 402-405 Mhz for communication with IMDs.

[1] `http://iml.jou.ufl.edu/projects/fall02/susi/tylenol.htm`
[2] `http://www.wired.com/politics/security/news/2008/03/epilepsy`

## Why are IMDs important

IMDs are important because they can improve the quality and length of people's lives. We expect to see that an increasing number of citizens will be in need of healthcare in the future. We expect this, because the European citizens are aging [68] and in general older people are more in need of healthcare than younger people [10]. People are attached to their homes and prefer to live in a familiar environment as long as possible [26]. But with many citizens in need for healthcare, all preferring to stay at home, we need to optimize the allocation of healthcare professional resources. A way to contribute, is to reduce the traffic time of the healthcare professionals [69]. This is possible by transmitting medical data over a longer distance. IMDs often come with external equipment which is capable of this. Therefore, it possible for a healthcare professional to view measurement data remotely and explain it to the patient without being physically close to him or her.

## Introduction to embedded security

Before we continue it is important to define some basic security concepts. Defining them in this section ensures a common ground on terminology. A reader with expertise in the area of information security could skip this section and only use it when he or she suspects ambiguity in terms. An *embedded system* is defined by the Embedded System Institute as[3]: "a combination of hardware and software components that are embedded into a product or application to allow it to interact intelligently with its environment". A *vulnerability* is a weakness in a product that could be triggered or exploited by a threat agent. An information security *threat* is an event or danger that might adversely affect an asset. The damage may be caused by exploitation of a vulnerability by a treat agent, sometimes called attacker. ISO 31000:2009 defines *risk* as "the effect of uncertainty on objectives". It is often calculated by multiplying the estimated probability of an event with the impact of that particular event. The CIA triad is an abbreviation for confidentiality, integrity and availability and is used to reason about security requirements for (information) systems. ISO/IEC 27001:2005 defines *Confidentiality* as: "the property that information is not made available or disclosed to unauthorized individuals, entities, or processes" ISO/IEC 27001:2005 defines *Integrity* as: "The property of safeguarding the accuracy and completeness of assets". ISO/IEC 27001:2005 defines *Availability* as: "the property of being accessible and usable upon demand by an authorized entity". A *penetration test* is a test in which a penetration tester legally simulates an attack on a system as if it were attacked by a real attacker. The goal of this test is to improve the security of the system. A *penetration test standard* is the description of a procedure to perform a penetration test. An *open standard* is a publicly free available standard.

## Why should we worry about IMD security

IMD security attracts media attention[4][5][6][7][8]. From the moment that security researchers started to explore the field of IMD security, notable results have been published. In 2008 security researchers discovered vulnerabilities in a pacemaker [38], in 2011 security researchers found a security vulnerability in an insulin pump [62] and in 2012 more vulnerabilities in pacemakers where found [42]. In all cases, the researchers where able to modify (therapy) settings from the IMD, without being an authorized party. If a malicious person exploits one of these vulnerabilities he or she could do serious harm to the patient. In the case of the pacemaker, the attacker could change the therapy settings to a hearth rhythm frequency which is possibly harmful or lethal to the patient. In the case of the insulin pump the attacker could insert a bolus dose (a doses of fast acting insulin)

[3] http://www.esi.nl/research/embeddedsystems.dot
[4] http://www.economist.com/node/21556098/
[5] http://www.wired.com/gadgetlab/2008/03/scientists-demo/
[6] http://www.wired.com/threatlevel/2011/08/medical-device-security/
[7] http://www.wired.com/threatlevel/2012/04/security-of-medical-devices/
[8] http://www.washingtonpost.com/investigations/health-care-sector-vulnerable-to-hackers-researchers-say/2012/12/25/72933598-3e50-11e2-ae43-cf491b837f7b_story_2.html

into the patient, which may lead to hypoglycemia and may therefore be harmful or lethal to the patient. We should worry about IMD security because we worry about the patient his or her safety. To improve IMD security we want to contribute to the field of IMD security assessments methodologies. Therefore, our main question for this research is:

**Main Question.** How can we improve IMD security?

## 1.1. Organisation

To find an answer to this problem, we divide the problem in five research questions, which we try to answer in this thesis. We start this thesis by discussing eight relevant papers about IMD security in Section 2. Followed, in Section 3 by a discussion about three IMD architectures: a pacemaker, an insulin pump and a deep brain stimulator. Based on these three IMD architectures we define one abstract IMD architecture and enumerate the common characteristics for it. We then discuss the architectures, threats and vulnerabilities of a SCADA system and a (Web) Information system and compare the characteristics of these systems to the characteristics we found for IMDs. Based on this information we make a list with desired security properties for an IMD.

**Research Question 1.** What security properties are desirable for an IMD?

After discussing the similarities and characteristics between IMDs, SCADA systems and (Web) information systems we focus on vulnerabilities and attacks. In Section 4 we discuss ten vulnerability types which may be applicable to an IMD. For these vulnerabilities we identify what type of attacker is able to exploit them and what the possible impact of successful exploitation might be.

**Research Question 2.** How can we attack an IMD?

With the information about the architecture, characteristics, differences and security properties we desire, we can evaluate the applicability of the OWASP verification standards and the 20 CSC v4.0 from SANS. We end Section 5 by discussing the relevant parts of ISO/IEC 27001:2005.

**Research Question 3.** How applicable are the current methodologies for IMD security testing?

In Section 6 we test the above theories in practice by performing a security assessment on the⋆⋆⋆⋆⋆. We will assess the IMD to find out, to what extent the IMD satisfies the desired security properties we defined in Section 3. By evaluating publicly available documentation, interviewing healthcare professionals and attacking the IMD we evaluate the weaknesses of the system.

**Research Question 4.** What is the current status of IMD security?

Based on previous sections we propose a security assessment methodology for IMDs. In Section 7, we discuss the necessary steps to plan, execute, report and evaluate the security of an IMD. Because we do not think our IMD security assessment is the only possible way to improve IMD security, we will recommend several security improvements in Section 8. We order these improvements on three levels: The IMD itself, The healthcare professional equipment and the whole healthcare equipment infrastructure.

**Research Question 5.** Can we mitigate IMD security risks?

Finally, in Section 9, we give an overview of the answers to the research questions as stated above and answer the main question.

# 2. Related work

This section provides an overview of the related work in the field of IMD security. It will cover papers about IMD security as well as aspects from the security of SCADA systems. SCADA systems have several similarities with IMDs, since there is more research available on these systems it is useful to consider them as well.

### Approach for the Literature study

In this literature survey we focus on research question four: "What is the status of IMD security?" For the literature study we selected all relevant results from the TU/e academic search engine[9].

### Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defences [38]

In their paper Halperin et al. [38] investigate the security of implantable cardiac defibrillators (ICD). The version and manufacturer of the device is not disclosed in the paper but the 175 kHz frequency range is revealed. By reverse engineering they discovered that radio transmissions were executed with a frequency shift keying modulation scheme. By analysing the signals and comparing them with the known plain text they discovered that both the IMD and the programmer encode the communication in Non-Return-to-Zero inverted with bit stuffing. By knowing these modulation scheme's they had the basis to start attacking the pacemaker. Halperin et al. [38] where able to record the signal for IMD identification and the signal for interrogation. After replaying the signal they received the same data as if the programmer was requesting it. The researchers demonstrated that they were able to modify IMD patient data and therapy settings. Finally, they proposed three zero power defence mechanisms for IMDs.

### Hijacking an Insulin Pump: security attacks and defences for diabetes therapy system [62]

In this paper Li, Raghunathan and Jha [62] describe a method for hacking an insulin delivery system operating on 915MHz. The Universal Software Radio Peripheral (USRP) is a tool capable of intercepting and replaying radio signals. We discuss this tool in subsection 3.5. By using the USRP Li, Raghunathan and Jha [62] where able to intercept communication and replay (modified) communication. They were also able to jam the communication channel. One remarkable achievement was the ability to inject a bolus dose. A bolus dose is a fast acting dose of insulin directly inserted into the human body. An inadequate dose may lead to hypoglycaemia and endanger the patient life. The paper ends with security recommendations such as the use of rolling codes for authentication to prevent replay attacks.

### Security and privacy for Implantable Medical Devices [39]

In this paper, Halperin et al. [39] discuss several important security properties for implantable medical devices such as data properties, safety versus security properties, usability properties and privacy properties. They classify adversaries in four classes: passive adversaries, active adversaries, coordinated adversaries and insiders. In Section 4 we explain and extend this set of attackers. They make a clear distinction between adversaries with standard commercial available equipment and custom, home made equipment. The paper concludes with security recommendations, such as a second channel notification (audio signal) when a connection is established..

### Encryption on the Air: Non-Invasive Security for Implantable Medical Devices [6]

In his master thesis, Al-Hassanieh, Electrical Engineering and Science [6] uses an implantable cardiac defibrillator, a pacemaker programmer and the USRP2 software radio boards to experiment with an IMD shield. An IMD shield is an external device which jams unauthorized communication

[9] `http://w3.tue.nl/nl/diensten/bib/over/bibliotheeklocaties/wi/zoeksystemen`

from the IMD. When somebody is successfully authenticated to the shield, the jamming stops and the healthcare professional could connect to the IMD. Since the IMD shield is external it is easy to replace the battery or whole device. Therefore it becomes feasible to implement authentication and cryptography for the IMD.

#### Patients, pacemakers, and Implantable Defibrillators: Human Values and Security for Wireless Implantable Medical Devices [23]

In this paper Denning et al. [23] interviewed 13 individuals with an IMD. There goal was to to explore the view of the patients regarding IMD security. They evaluated four options about how and where to modify an IMD to improve its security: passwords, additional patient body modifications, patient behaviour changes and a passive option. The passwords are evaluated together with the "additional patient body modifications" option. The researchers define body modification as: visible and invisible tattoos placed on the patient body. To ensure the healthcare professional can always access the IMD via a password, even when the patient is unconscious they asked the patients how they feel about body modifications. 55% of the interviewee did not like this solution. For the second method a cloak is considered as a "Patient behaviour" solution. From the description of the wristband / cloak we assumed that the researchers refers to the research of Halperin et al. [38]. 45% the patients preferred this solution. The "passive method" changes the access policy of the IMD when an emergency situation is detected. 27% of the patients preferred this method. We think that 13 individuals is not significant enough to draw a full conclusion about the preference of IMD patients.

#### Take Two software updates and see me in the morning: the case for software security evaluations of medical devices [40]

In this paper Hanna et al. [40] describe four different types of application weaknesses in Automated External Defibrillators (AEDs). The AED is an external medical device but follows similar procedures as the implantable medical device. Therefore we deem this research applicable for our research. By fuzzing the update procedure of the AED, Hanna et al. [40] discovered several ways to crash the program. From the generated crash reports they concluded that the system is vulnerable for integer and buffer overflow attacks. By populating the verification table, the cyclic redundancy check (CRC) could be manipulated in such a way that a modified firmware image is accepted by the system. The second flaw they found is in the password authentication scheme. Because the password file is stored locally and everybody has the right to read and update the file, it is easy to fully control the device. Finally, they discovered that the credentials to connect to the remote FTP server were transmitted in plain text.

#### Analysis of unencrypted and encrypted wireless keyboard transmission implemented in GNU Radio based Software-Defined Radio [29]

In their master project Fähnle and Hauff [29] describe a method for hacking an Wireless Keyboard at the 27 MHz frequency by using the Universal Software Radio Peripheral (USRP). We will discuss the USRP in Section 3.5. By capturing and replaying radio packets Fähnle and Hauff were able to hack wireless keyboards. Because most IMDs also communicate over Radio Frequency, capturing and replaying of radio packets with the USRP may also be an effective attack to them.

#### The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks [80]

In 2000, Stajano and Anderson [80] described the sleep deprivation torture attack. This attack focuses on the availability of a portable / implantable device. The sleep deprivation torture attack works by trying to make as much connections / key exchanges to a device as possible. This could result in battery exhaustion. For an IMD, accelerated battery exhaustion implies early battery replacement and therefore unnecessary surgery. Because battery exhaustion is a risk in IMD security we deem this paper applicable for our research.

# 3. IMDs and their architecture

In this section we discuss the architecture and desired security properties for IMDs. We start by discussing the architecture of three IMDs: a pacemaker, a generic insulin infusion pump and an implantable deep brain stimulator . Based on those three architectures, we define an abstract IMD. We use this abstract IMD to reason about IMD properties, rather than a specific pacemaker, insulin pump or deep brain implant. Since we want to evaluate the applicability of penetration test standards of other systems, in Section 5 we also discuss the architecture of a web based information system and the architecture of a SCADA system. Based on these architectures we discuss the characteristics of IMDs in comparison to the other systems. We end this section by enumeration of the desired security properties for IMDs.

## 3.1. Architecture of an IMD

To talk about IMDs and their security it is necessary to give an abstract overview of the architecture of an IMD. Unfortunately, as we will further discuss in subsection 6.1 most manufacturers chose not to cooperate with this research. We also did not find publicly available software or hardware architecture documentation. Therefore we try to reconstruct high level architectures by other publicly available information such as customer or healthcare professional manuals, academic literature and interviews. After discussing the architectures, we propose an abstract architecture on which we base our IMD security properties.

### Architecture of a pacemaker / defibrillator

A pacemaker is an IMD used to monitor and stimulate the patients heartbeat. To stimulate the proper functionality of the heart the pacemaker has the ability to give an electronic pulse to the heart [92, 90]. The pacemaker system consists of three devices: the pacemaker, the ⋆⋆⋆⋆⋆ programmer and the⋆⋆⋆⋆⋆. The ⋆⋆⋆⋆⋆ programmer and ⋆⋆⋆⋆⋆ both communicate with the


Figure 1: Pacemaker hardware ⋆⋆⋆⋆⋆

pacemaker. The ⋆⋆⋆⋆⋆ and the programmer device do not communicate with each other and may not communicate simultaneously with the pacemaker [1]. To activate communication between the pacemaker and the programmer device it is necessary to supply a burst from the programmer device head. The ⋆⋆⋆⋆⋆ manual[10] describes a procedure for a less privileged connection which does not need a burst by the programmer head button. In Figure 2 one could observe that this burst is a physical activation of the electronic circuit by magnetism [91]. In Figure 3 and as described by Warren et al. [90] the pacemaker architecture has three important subsystems: the sensor, the shock leads and the control parts.

1. **The sensor** acquires the electrocardiogram (ECG) input.

[10] ⋆⋆⋆⋆⋆

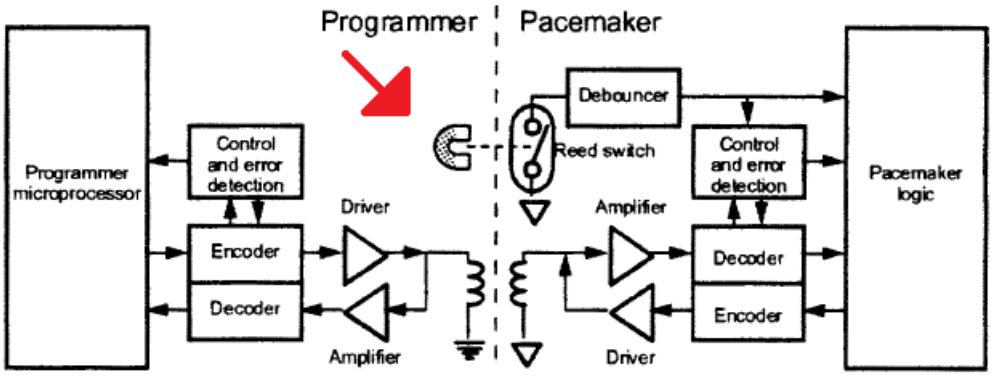



Figure 2: Pacemaker programmer communication interface [91]

2. **The control part** process the data acquired by the sensors. Based on the configuration, the logic and the sensor input the controlled decides what command to send to the output.

3. **The output** can intervene to protect the patients health, for example by giving an electronic pulse to the patients heart. Besides direct action some IMDs can also store (statistical) data for later analysis by a healthcare professional.

In Figure 3 the architecture also includes the battery, bus and other electronic components. However, one could easily distinguish the three important subsystems as described above. After the

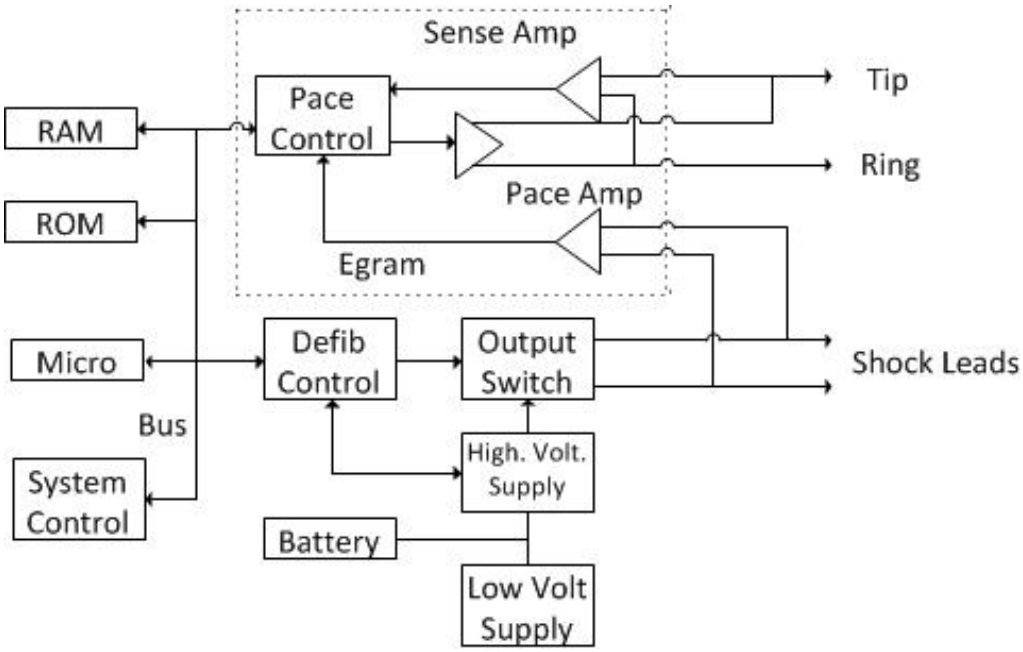



Figure 3: Pacemaker system architecture [90]

startup phase either by burst or a high frequency signal, the hardware establishes a radio connection in the 402MHz to 405MHz band.

### Architecture of a generic insulin infusion pump (GIIP)

An insulin infusion pump is an IMD that monitors the blood glucose levels from a patient. Based on that information the insulin pump could insert insulin to adjust the blood glucose level to a suitable level. In their paper Zhang, Jones and Jetley [96] describe an abstract design model for insulin infusion pumps. We could derive four important components from the model in Figure 4: the pump controller, the user interface, the infusion set and the environment. As shown in Figure 4

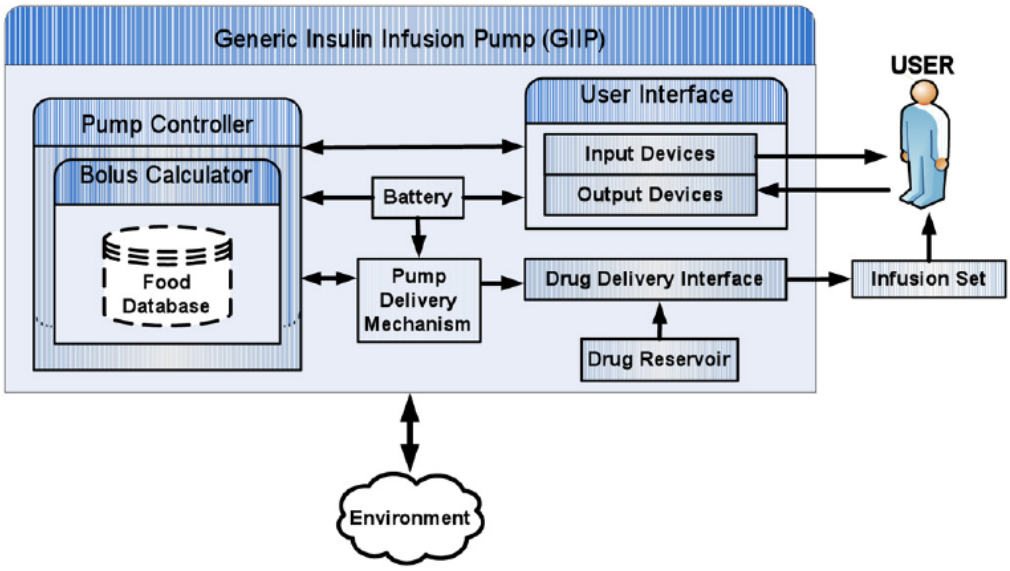



Figure 4: System architecture of generic insulin infusion pump [96]

the pump controller receives input from two sources: the environment and the user interface. For the sake of simplicity, we generalize this to one component: the input.

1. **The input** is divided in two parts: the sensor input and the therapy input. The sensors are placed in a different device known as the glucose meter [34]. The therapy input makes it possible for the healthcare professional and patient to adjust the therapy settings.

2. **The controller** interprets the data and based on the configuration and logic sends a command to the output.

3. **The output** or infusion set makes it possible to deliver an insulin dose to the patient which is determined by the controller.

### Architecture of a Bi-directional Brain-machine Interface for Deep Brain Stimulation

The Bi-directional Brain-machine Interface (BMI) for Deep Brain Stimulation (DBS) is an implantable medical device implanted in the human brain [81]. DBS works by stimulating specific brain regions with electronic pulses [57]. Although, it is still unclear how DBS really works, research has helped to clarify not only the neural mechanisms and targets that underlie the effects of DBS , but also the fundamental brain functions that are affected in the disorders for which DBS is used [57]. This therapy is effective for the treatment of Parkinson symptoms [61, 81], chronic pain, dystonia [57] and other neurological disorders [81].


Figure 5: DBS hardware [27]

As shown in Figure 5 also the DBS device (Figure 5 part one) is programmed by a healthcare professional via the programmer device (Figure 5 part three) [27]. The external hardware operated by the patient (Figure 5 part two) enables the patient to turn the device on or off, and to change the strength of the stimulation [27]. As show in Figure 6 the BMI has a high level architecture

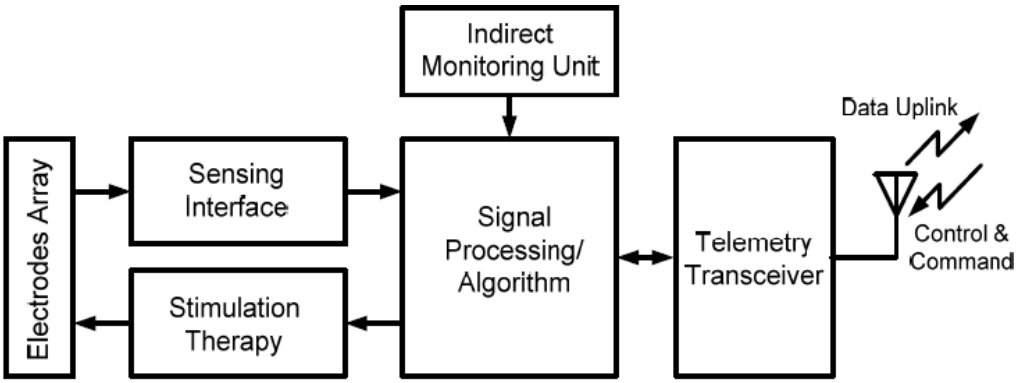



Figure 6: Function diagram for Implantable BMI [81]

similar to the pacemaker. Again, we distinguish three important components: the input, controller and output.

1. **Input** The BMI has a sensor for monitoring the neural activity. Besides the sensor input the BMI has an telemetry transceiver for input from the healthcare professional.
2. **The controller**, a signal processing algorithm takes the configuration from the telemetry and the signal from the sensors to determine the appropriate output [31].
3. **The output** could stimulate the Subthalamic Nucleus (the brain part assumed to regulate the decision threshold [61]) with a high frequency electrical signal .

#### Abstract architecture of an IMD

Before we can discussing the properties of a generic IMD it is necessary to introduce an abstract model of an IMD. Without an abstract IMD we can only talk about the pacemaker or the insulin pump. But, with an abstract IMD we can talk about the generic characteristics and general security properties of IMDs. As seen in the above architectures, the three IMDs we have described all have components in common. We modelled the components in Figure 7 and discuss them below. The first component is the data acquisition component which includes sensors for measuring. For example: hearth rhythm signals, neurological signals or blood glucose values. The second component is the telemetry control. The telemetry control makes it possible to connect wirelessly

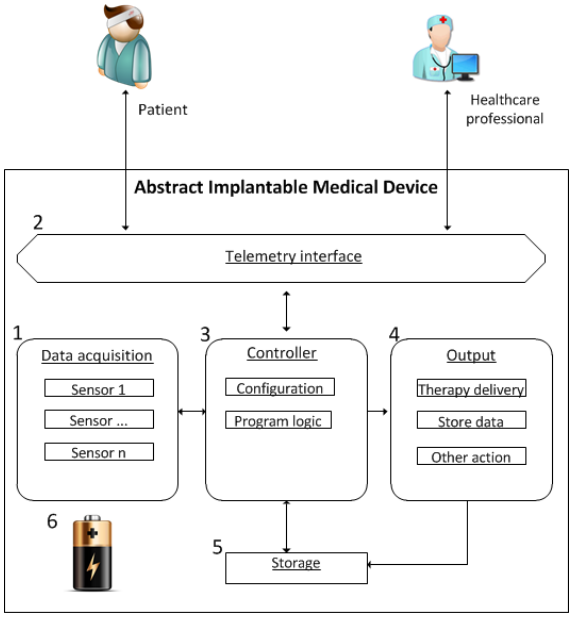



Figure 7: Abstract Implantable Medical Device

with external healthcare equipment. With this connection external equipment could program or configure the IMD. The third component is the controller component. This controller can be configured by a healthcare professional and contains, for example therapy parameters. The fourth component is the action component. The action component can delivers therapy based the configuration and sensor input or store data for later usage. We did not model the data storage as a database because not all IMDs [81], as we discussed above had a real database component. Therefore we decided to abstract it to the level of data storage (component five). Finally, we modelled a disconnected battery (component six) in the abstract IMD to emphasize that an IMD only has an internal power source.

## 3.2. Architecture of other systems

In Section 5, we want to evaluate the applicability of penetration test standards from other systems. We elaborate on the justifications of the use of these systems and standards in the introduction of Section 5. We now discuss the architecture of a web based information system and the architecture of a SCADA system. At the end of each architecture discussion, we enumerate the threats and vulnerabilities for the specific system. For the sake of simplicity we assume the same set of attackers for all systems on which we elaborate in subsection 4.3.

### Architecture of a SCADA system

A Supervisory Control and Data Acquisition (SCADA) system provides control and monitoring abilities for mechanical and electrical utility systems [88]. A SCADA system is just a flavour of the many types of control systems. There are multiple architectures and implementations for control systems. But they all have in common that they are originally designed to be able to operate in an isolated environment [52]. The system could be managed and operated via a Human Machine Interface (HMI). The relation between the HMI as shown in Figure 8 and the Master Terminal Unit (MTU) can be compared to the relation between the pacemaker programmer and the pacemaker. In this case the HMI connects to the MTU which is able to connect to the Programmable Logic Controllers (PLCs) or Intelligent Electronic Devices (IEDs). Via this connection the operator could configure or reprogram the PLCs / IEDs. Since this connection can be wireless for some

PLCs / IEDs, the connection is comparable to the connection between the IMD and the IMD programmer. The PLC or IED could be connected to sensors and acquire sensor data. Based on the settings of the PLC / IED and how the PLC / IED is programmed the PLC / IED makes a decision. An example of such a decision is to invoke a physical action via the actuators or to store data in the historian. The historian as shown in Figure 8 at the "controller system" level keeps track of historical data. Although the IMD data storage is far more limited, the historian at the "controller system" level is comparable to the data storage inside the IMD. The historian and historian database on the supervision level are, according to the above analogy more comparable the electronic health record system. We divide the "control system" part from Figure 8 into three important controls: the data acquisition component, the controller process and the actuators.

1. **The input** is acquired by sensors on the SCADA system.
2. **The controller process** PLC or IED, processes the input from the sensors based on the logic and parameters programmed by the operators via the MTU.
3. **The output** also known as actuators are the components that perform the actual supervisory.

By reviewing the literature [88, 86, 70, 98, 53, 45], we identified a list of threats and vulnerabilities which are applicable for SCADA systems. We summarize the vulnerabilities in: Table 2 and the threats in: Table 1.

| ID | Threat | Description |
|---|---|---|
| $T_1$ | Unauthorized access / command sending [88] | Allows an attacker to control the system. |
| $T_2$ | Unauthorized software / data modification | Allows an attacker to control the software and data. |
| $T_3$ | Denial of Service [88] | The system is made unavailable. |
| $T_4$ | Information leakage [88] | Confidential information from the SCADA system leaks to the public. |
| $T_5$ | $\text{Repudiation}_1$ | The attacker uses the system as a proxy server to attack other systems. |
| $T_6$ | $\text{Repudiation}_2$ | The attacker uses the system as a stepping stone to attack the internal network. |
| $T_7$ | Resource scavenging | The attackers uses the machine resources for its own benefits (for example the CPU for password cracking). |
| $T_8$ | Impersonation attacks | Sending false information impersonating the system. |

Table 1: Potential threats to SCADA systems.

| ID | Vulnerability | Description |
|---|---|---|
| $V_1$ | Unencrypted communication | The attacker could read and modify the data transmitted between the SCADA system and the control room. |
| $V_2$ | Weak encryption | If the encryption is weak, the attacker attempt to crack it offline. |
| $V_3$ | Authentication protocol vulnerabilities | An attacker could exploit several authentication weaknesses to obtain unauthorized access to the system. |
| $V_4$ | Software Vulnerabilities | Software vulnerabilities include the specific software vulnerabilities of the SCADA system. |
| $V_5$ | Non redundant Power Supply | A SCADA system may not have its own power supply and therefore be dependable on the national power grid. If this grid fails or the power cables to the SCADA system fail than it may not function any longer. |
| $V_6$ | Unsecured Physical access | An attacker may have physical access to the SCADA system. This allows the attacker to add, remove or modify hardware. |
| $V_7$ | Untrained / Unaware employees | Social engineering from an information security perspective is the art of hacking a system trough the people who operate them. |
| $V_8$ | Temperature / conditions | The system may be vulnerable for high or low temperatures caused by the weather or an accident like fire. Other environmental vulnerabilities could be dust or moisture. |

Table 2: Vulnerabilities of SCADA systems.

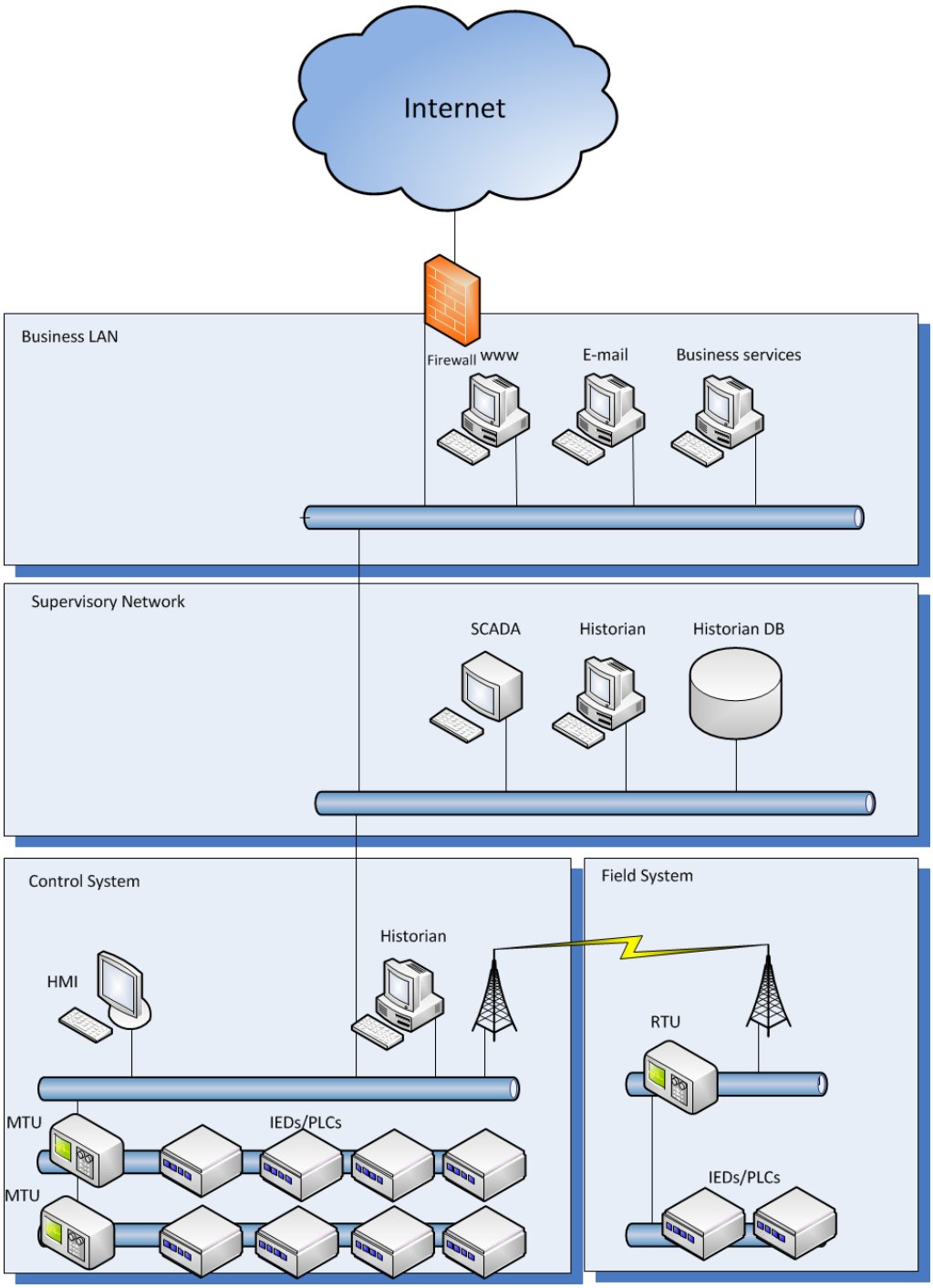



Figure 8: SCADA system [55, 22]

### Architecture of an web based information system

A web information system is a system which represents, stores, distributes and allows to modify information [89, 76]. The information is often stored in a database and could be accessed via a web application [76]. The web application is mostly accessible for all users via the public internet [89]. We identified three important components: the client, the web interface / application and

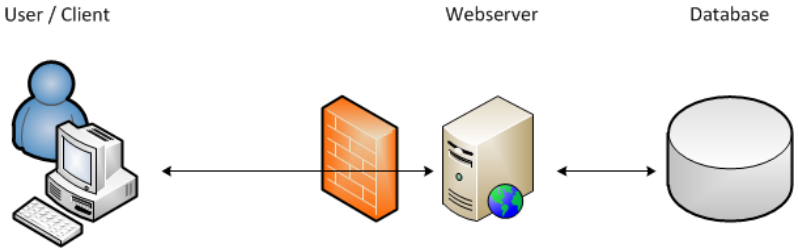



Figure 9: Web information system architecture

the database.

1. **The client** makes a request to the web interface / application.
2. **The Web interface / application** Sends the data and commands back to client. If a command for reading, adding or modifying data is given by the client, the web application processes this command.
3. **The databases** stores data and processes requests for adding or modifying [76].

The client is often a web-browser [77, 76] controlled by the end user but can also be another (web) application. The relation between the client and the web interface can be compared to the relation between the IMD programmer and the IMD. This relation is weaker than the comparison of the HMI / MTU relation to the pacemaker programmer / pacemaker relation because web applications are often accessible for all users [89]. Although the IMD data storage is far more limited, the database is comparable to the data storage inside the IMD. A difference is that not every web application has a database. Another difference is that the web database may also be used by other (web) applications but that the IMD storage should be bounded [38, 34, 71, 43] to IMD usage only. The similarity is that both systems are capable of reading, writing and modifying data [38, 34, 71, 43, 76]. The web application is often programmed based on the language provided by the webserver [76]. The webserver [33] deals with the details as memory allocation and the protocol level. Examples of webservers are Apache HTTP Server Project[11] and Internet Information Services (IIS)[12] By reviewing literature [48, 49, 97, 16], we identified a list of threats and vulnerabilities which are applicable for (web) Information systems. We summarize the vulnerabilities in: Table 3 and the threats in: Table 4.

[11] `http://httpd.apache.org/`
[12] `http://www.iis.net/`

| ID | Vulnerability | Description |
|---|---|---|
| $V_1$ | Unencrypted communication | The attacker could read the data transmitted between the web information system and the client. |
| $V_2$ | Weak encryption | If the encryption is weak, the attacker could capture the communication and store it for offline cracking. |
| $V_3$ | Authentication protocol vulnerabilities | An attacker could exploit several authentication weaknesses to obtain unauthorized access to the system. |
| $V_4$ | Software Vulnerabilities | Software vulnerabilities are vulnerabilities specific for the software of the (web) Information system. |
| $V_5$ | Unsecured Physical access | An attacker may have physical access to the (web) Information system. This allows the attacker to add, remove or modify hardware. |
| $V_6$ | Untrained / Unaware employees | Social engineering from an information security perspective is the art of hacking a system trough the people who operate them. |
| $V_7$ | $3rd$ party / vendor vulnerabilities | Since most web applications are based on a web server which is based on the operating system the application inherits all vulnerabilities in these systems. |
| $V_8$ | Weak configuration | During the configuration phase the installer makes some mistakes. For example the assignment of more privileges than necessary, setting of weak passwords or he forgets to remove the installation files after the installation. |
| $V_9$ | Typosquatting | The attacker registers a domain name which is very similar to the real name but differs one character which could be a common typing mistake (p instead of o). This domain could then be used as a platform for phishing etc. |

Table 3: Vulnerabilities of (web) information systems.

| ID | Threat | Description |
|---|---|---|
| $T_1$ | Unauthorized access / command sending | Allows an attacker to control the system. |
| $T_2$ | Unauthorized software / data modification | Allows an attacker to control the software and data. |
| $T_3$ | Denial of Service | The (web) Information system system is made unavailable. |
| $T_4$ | Information leakage | Confidential information from the (web) Information system leaks to the public. |
| $T_5$ | $\text{Repudiation}_1$ | The attacker uses the system as a proxy server to attack other external systems. |
| $T_6$ | $\text{Repudiation}_2$ | The attacker uses the system as a stepping stone to attack other systems on the internal network of the victim. |
| $T_7$ | Resource scavenging | The attackers uses the machine resources for its own benefits (for example the CPU for password cracking). |
| $T_8$ | Impersonation attacks | Sending false / wrong information impersonating the (sensors of the) system. |
| $T_9$ | Defacement | This attack focuses on the presentation of the data via the (web) server. Instead of displaying the normal web page it could display any content the attackers wants to show including illegal content. |

Table 4: Threats to (web) Information systems.

## 3.3. IMDs compared to other systems

In this section we compare IMD characteristic to (web) information system characteristics [52] and control system characteristics [52].

| | (Web) Information system | Control System | IMD |
|---|---|---|---|
| **Power supply** | National power grid | National power grid | Non rechargeable battery |
| **Real time** | No | Yes | Yes |
| **Interaction** | Often via internet | Often via internet | Wireless |
| **Processing** | Multi threading | Sequential | Sequential |
| **Replaceable** | System is highly replaceable | Replacement is expensive | Replacing is a risk for the patient |
| **Users** | Multiple users and groups supported | Limited users and groups supported | Only three actors |
| **Dedicated to single task** | No | Yes | Yes |
| **Storage space** | Limited to owners resources | Limited to owners resources | Very limited |
| **Processing power** | High | High | Very limited |
| **Size** | Variable | Variable | Around 10 cm |
| **Cryptographic capabilities** | Full | Full | Very limited |
| **Environmental dependencies** | Very dependent | Not that dependent | Not that depended |
| **Reusability** | Easy to reuse | Reusable | Hard to reuse |

Table 5: Comparison of IMD characteristics

#### Power supply

Both, (web) information systems and SCADA systems are powered by the national power grid. Although they may differ in voltage and power consumption they are similar in the sense that the power supply is continuous and replaceable. If these systems are very critical for society, business or individuals they may even have an independent backup power supply. IMDs do not have this luxury. Since IMDs are implanted in the human body they cannot connect to the national power grid. The IMD is depending on a battery which is currently not able to recharge from inside the body. The replacement of an IMD or IMD battery implies surgery [43]. Surgery comes with the risk of infection or death [36]. Therefore, most IMDs are programmed to be as energy efficient as possible [38, 34, 43, 69, 71, 37]. As shown in Section 3.3 there are IMDs which consist for 50% of battery capacity.

#### Real-time or reaction based

A (web) information system usually waits with performing an action until a client requests an action [52]. SCADA-systems [52] and IMDs collect data in real time [62, 81, 92] and analyse the data to react on the environment.

#### Interaction

Most (web) information systems are accessible via the internet and accept connections from every user who wants to connect to them [47]. Originally, SCADA systems where designed to be able

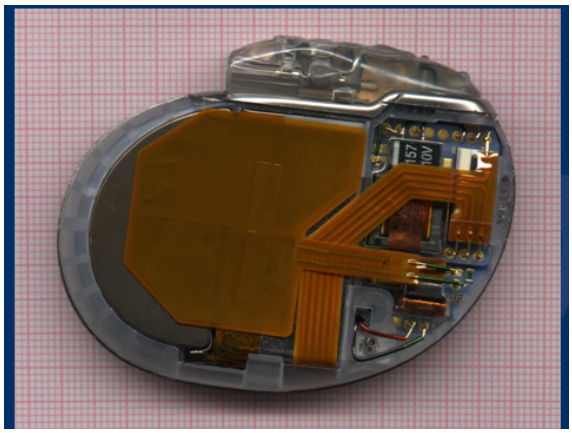


Figure 10: Limited battery (left) capacity of an ICD

to operate in an isolated environment [52]. If there were connections, they were mostly LAN connections. This is not the case any longer. Nowadays an increasing number of SCADA systems is available via internet by a web-interface [7]. Most IMDs use short range wireless communication to transmit their data to a computer or additional healthcare equipment [43]. This computer or additional healthcare equipment is often able to transmit the data via internet to the hospital [34, 2, 82]. So, all three systems have interactions with other systems or with end users but they differ in communication technology and the amount of interaction they have.

### Processing

A (web) information system could process multiple requests from multiple users in parallel. This is called multi threading. Control systems on the other hand, usually processes one request at a time [52]. In fact, it should not be possible, to process two opposite commands (for example open and close) at the same time. Therefore we characterise the processing from a SCADA system as sequential processing [52]. The same holds for the IMD, it should not be possible to process two opposite commands at the same time. There may be an exception of this rule during the data transmission phase. The IMD should run a parallel process because the main process (improving the patient health) should not be interrupted because of a data transmission.

### Replacement

A (web) information system is highly replaceable. Sometimes, (web) information systems are redundant and connected by a load balancer. In this case a (web) information system can be replaced without effecting availability. SCADA system are often customized hardware systems. Therefore the replacement of a SCADA system is usually more expensive than replacing a (web) information system. IMDs are implanted in the human body and replacing an IMD implies that the patient needs surgery. Because the surgery is a risk for the patients health [36] and IMDs are expensive we classify the IMD as hard to replace.

### Users

Most (web) information systems have multiple users and groups. SCADA-systems have limited users and groups. IMDs should only have three actor types: healthcare professionals, the patient and the IMD developer [34].

#### Dedication to task

A (web) information system is designed to be highly expendable and modifiable. One could easily add a web application to the web information system or modify an existing one. SCADA systems and IMDs are designed for a specific task.

#### Storage space

Both, (web) information systems and the SCADA systems could be connected to the internet. They have the ability to store data on an external source but they have to be able to work in an isolated environment. We deem their storage space only limited to the owners will to purchase storage space. The IMD in contrary does often not have a permanent connection and since it is designed to be as small as possible it does not have much room for temporary storage. At least one IMD, which was still in use around 2011 had only 8 KB storage space [43]. Therefore we deem the storage space of an IMD very limited [38, 34, 71].

#### Processing power

Both, (web) information systems and the SCADA systems are connected via the internet. They have the ability to use external processing power if necessary but they have to be able to work in an isolated environment. Therefore we deem their processing power unlimited. IMDs however, are very limited in processing power because they have to fit inside the human body. Since they cannot connect to an external processing source at any desired moment we rate the processing power of an IMD as very limited [71, 38, 34].

#### Size

Webserver could vary size from a credit card sized Raspberrypi[13](8.5cm) up to a closet size server. SCADA systems could range from a small Intelligent Electronic Device (IED) up to big water or nuclear control systems. Because IMDs have to fit inside the human body they are as small as possible. During our IMD study we did not find any IMD bigger than 10 cm.

#### Cryptographic capabilities

Since both, (web) information systems and SCADA systems have in potential unlimited processing power and storage capacity they have the full capabilities to use cryptographic technologies. IMDs however have to be as energy efficient as possible. Therefore they should be very limited with the use of power consuming cryptography [38, 62].

#### Environmental dependencies

Often, (web) information systems are found in locations which are dedicated for their use, for example server parks. Therefore we rate them very depended of the environment. SCADA systems operate in many different environments. Some SCADA systems operate outside in nature and must be resistant against dust, water and over heating. Since they where originally designed to be able to operate in an isolated environment we rate them: minimal depended. IMDs operate in a human body which is bound to average conditions about humidity, temperature and movement. As these conditions are violated it is more likely that the patient is already dead. Therefore we rate them not that depended on the environment.

#### Reusability

If a (web) information system is not needed for a specific task any more and it still works and meets the requirements for a new project then it could be easily reused. Therefore we rate this system

[13] urlhttp://www.raspberrypi.org/faqs

very reusable. SCADA systems are often more dedicated to a task. However, many machinery tasks are needed on more than one place. If the SCADA systems still fit the requirements they could be reused. Since IMDs are inside a human, new patients may be reluctant in using an old IMD for several reasons. Besides the will of the patient also legal and medical reasons my play a role in the decision not to reuse IMDs. It is however possible and healthcare professionals are experimenting with it [75]. We therefore rate it: hard to reuse.

#### Similarity

By comparing the characteristics, we have seen that there are similarities between IMDs and SCADA systems. We have also seen that there are similarities between IMDs and (Web) information systems. All have the ability to process and store data. But, the IMD is more limited in doing so than the other systems. This is because of two fundamental differences. The IMD is depending on a limited power source, a battery. It is not possible to have easy physical access to the IMD or its battery. Based on the descriptions above we found that SCADA systems have five characteristics which are similar to IMDs and seven characteristics for which the concept is similar but the application in the IMD is more limited. For (web) information system we found one characteristic that is similar to IMDs and five characteristics for which the concept is similar but the application in the IMD is more limited.

## 3.4. Use of radio frequency

The European committee reserved the frequency range from 402-405 MHz for Medical Implant Communication Systems (MICSs) [19]. Therefore we give a short introduction to the physics needed to understand radio signals. We describe three demodulation scheme's needed to recognise the right scheme when a security penetration tester is conducting a security test. Finally, we give a short introduction to a hardware device to work with radio signals and software to process it.

#### Introduction to waves

We now introduce some basic terminology about waves, based on Calculus [5] (page 206-207) Radio communication works by osculation. This osculation is often described by the general solution of the auxiliary simple harmonic motion equation based on time (t):

$$y(t) = A \cdot cos(\omega(t + t_0)) + B \cdot sin(\omega(t + t_0)) \tag{1}$$

The *period* of the curve (T) is the time interval between two consecutive osculations. It is measured in Hertz, where one Hertz (1 Hz) is one oscillation in a second. The quantity $\omega = \frac{2\Pi}{T}$ is the circular frequency or *angular frequency* and is measured in radians per second, 1 cycle = 1 revolution = $2\Pi$ radians. The number $t_0$ is called the *time-shift*. The quantity related to the time-shift is called the *phase-shift* and is defined as $\omega \cdot t_0$. The *amplitude* is the difference between the top of the wave and the zero line as shown as y in Figure 11. Based on Equation 1 the amplitude (R) is defined as $\sqrt{A^2 + B^2}$ A *carrier wave* is often a sinusoidal wave which could be physically adjusted to transport an information signal (data) [79][59].

$$c(t) = A_c \cdot cos(2\Pi f_c \cdot t + \phi_0) \tag{2}$$

Where $A_c$ is the amplitude, $\phi_0$ is the starting phase and $f_c$ is the carrier frequency. [60] The process of adjusting the carrier wave is called *modulation* and focuses on adjusting the: frequency, amplitude or phase of the carrier wave [79][59][60]. A single adjustment to the carrier wave is called a *symbol* or *signal* [59] and a symbol could represent multiple bits. The changes or signal events made to the carrier per seconds is called the *symbol rate*. This is an important parameter because without synchronisation of this parameter between the sender device and the receiver it is impossible to reconstruct (demodulate) the original data stream. The technique for representing a continuous signal as a discrete signal is called *sampling*.

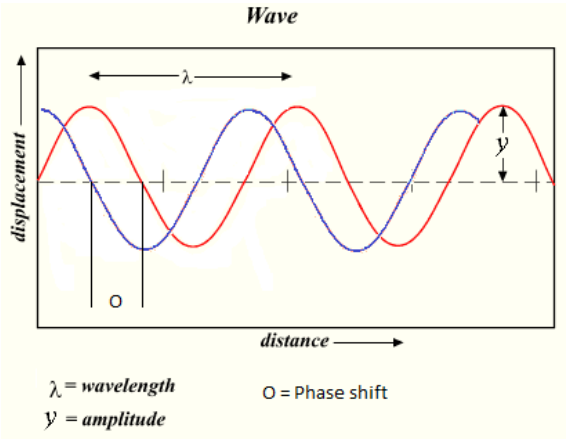



Figure 11: A wave with wavelength $\lambda$ and amplitude y

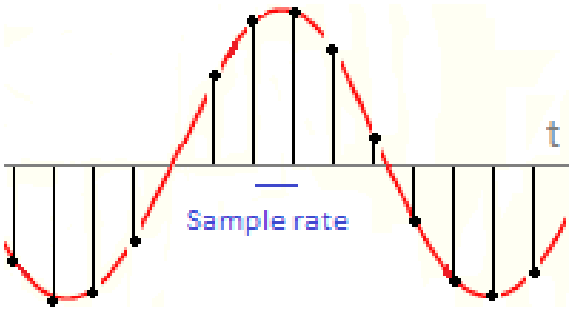



Figure 12: Sampling of a continuous signal

By storing the amplitude of the continuous signal by a fixed period of time, called the *sample rate* a continuous signal could be stored [78] on a discrete medium. To reconstruct a data stream transmitted via a continuous signal it is important to capture at least twice as many samples as the symbol rate. This is called the NyquistShannon sampling theorem [50]. An IMD transmits binary data [38] therefore we focus on the modulation of binary data instead of continuous signals. Figure 13 shows a binary message m(t) which we use as example message in the following

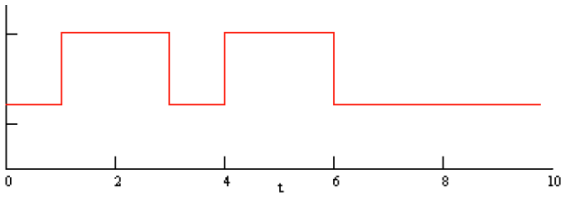



Figure 13: Binary message signal, m(t) [60]

modulation schema. The message represents the bitstream: 0110110000.

### Amplitude Modulation

Amplitude modulation is the multiplication of the amplitude of the carrier with the message signal [60].

$$s(t) = m(t) \cdot c(t) = A_c \cdot m(t) \cdot cos(2\Pi f_c \cdot t + \phi_0) \tag{3}$$

When working with digital communication the method is called *amplitude-shift keying* (ASK) Notice that the binary message in Figure 13 represents the zero bit as a non zero integer as message.

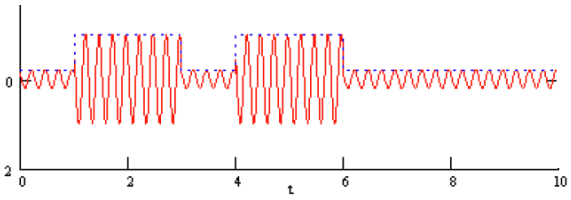



Figure 14: Carrierwave modulated with binary message (Figure 13) via Amplitude Modulation [60]

In the simplest form of amplitude-shift keying: *On-off keying* (OOK) the zero is represented as the absence of the amplitude. In more advanced amplitude shift keying schemes each amplitude level could represent another symbol.

### Frequency Modulation

Frequency modulation changes the frequency of the carrier with the message signal [60].

$$s(t) = \begin{cases} A_c \cdot cos(2\Pi f_1 \cdot t + \phi_0) & \text{for bit 1} \\ A_c \cdot cos(2\Pi f_2 \cdot t + \phi_0) & \text{for bit 0} \end{cases} \tag{4}$$

When working with digital communication the method is called: *Frequency shift keying* (FSK).

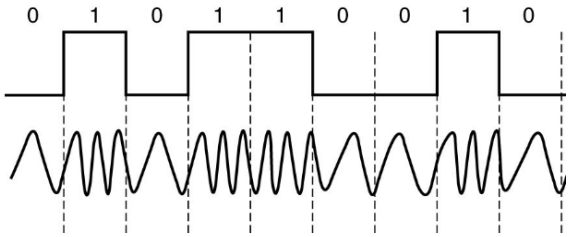



Figure 15: Carrierwave modulated with binary message by frequency modulation[14]

In this modulation scheme frequency one ($f_1$) represent the 0 bit and another frequency ($f_2$) represents the 1.

### Phase Modulation

Phase modulation changes the phase of the carrier signal to represent a symbol [60].

$$s(t) = \begin{cases} A_c \cdot cos(2\Pi f_c \cdot t + \phi_0) & \phi_0 = 0 \text{ for bit 1} \\ A_c \cdot cos(2\Pi f_c \cdot t + \phi_1) & \phi_1 = \pi \text{ for bit 0} \end{cases} \tag{5}$$

This is done by changing the phase to a function depended on time. For sending a binary message called *phase shift keying* this modulation is simple. The 0 is represented as itself and the 1 as $\pi$. For determining the demodulation scheme in the next section it is important to notice that the change in phase is very obvious when dealing with a discrete signal.

[14] `http://computing.dcu.ie/~humphrys/Notes/Networks/physical.phone.html`

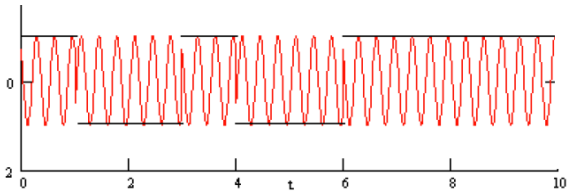



Figure 16: Carrierwave modulated with binary message (Figure 13) via Phase Modulation [60]

## 3.5. Sniffer equipment

In this section we discuss the hardware and software needed to perform a security assessment on an IMD. We start by the USRP hardware and via the concept of the Software defined radio and the USRP Hardware Driver (UHD) we finish with the GNU Radio software. For a graphical overview of all the components one could observe Figure 18 at the end of this section.

### USRP

The USRP1 is a Universal Software Radio Peripheral hardware (USRP) that provides entry-level RF processing capability[15]. The USRP is developed by Matt Ettus and sold via his company Ettus Research and its parent company National Instruments [11]. The USRP has a motherboard containing a Field Programmable Gate Array (FPGA), ADC(s),DAC(s), Programmable Gain Amplifier (PGA), an internal clock and an high-speed USB or Gigabit Ethernet link port to connect to a computer. The internal clock is specified as an 64 MHz crystal oscillator internal clock. This binds the maximum sample rate to 64.000. It is easy to expand the USRP by adding one or multiple daughterbords from the class: Receivers (Rx), Transmitters(Tx) or Transceivers(Tx and Rx). With all the daughterbords, the USRP is able to Transmit and Receive from 1 Mhz up to 4.4 GHz.


Figure 17: GNU Radio USRP (Universal Software Radio Peripheral)

### Software defined radio

For processing radio signals we need hardware to capture, store and analyse these signals. Normally embedded systems implement these operation inside the hardware [24]. This is expensive, time consuming and not flexible for our purpose. In a *software defined radio* components like: filters, amplifiers and modulators/demodulators are implemented in software [24]. Of course not every part of the radio can be software. The component converting an Analog signal to a digital one by sampling, called the *Analog to Digital Converter* (ADC) is still a hardware component. The component handling the other way around is called the *Digital to Analog Converter*(DAC). But since the (de)modulation scheme's are it is possible to test lots of different demodulation scheme's

[15] `https://www.ettus.com/product/details/USRP-PKG`

within a reasonable time. One SDR system which is widely used is the GNU radio [11] founded by Eric Blossom in 2001.

#### GNU radio and the GNU Radio Companion

GNU radio[16] is a free and open-source (licensed under GPL) software development toolkit that provides signal processing blocks for implementing SDR systems. These block based SDR systems are mainly written in python. For the basic systems the GNU Radio Companion could be used. GNU Radio Companion is a tool with graphical user interface which generates python code from a flow diagram[17]. The blocks itself which perform the performance-critical signal processing operations are implemented in C++ using, if available processor floating-point extensions.

In newer versions of GNU radio, GNU radio does not directly communicate with the USRP. The *USRP Hardware driver*(UHD)[18] is a host driver and API for future Ettus Research products. Making it possible to work with $3^{th}$ party software instead of only GNU radio.

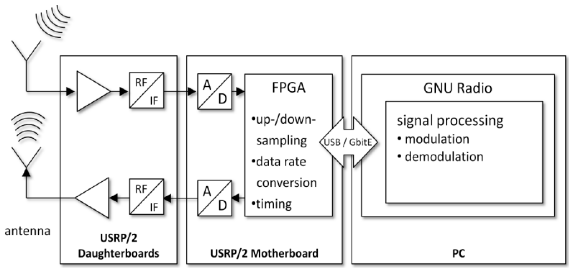



Figure 18: Architecture of the equipment [29]

## 3.6. Desired security properties

In this section we give an overview of security properties we desire for IMDs. Those security properties could be mutual exclusive, non existing or currently impossible to implement. But we stated them all because we find it important not to limit our desires on forehand. However, in practise one should choose which properties to satisfy and which not. The second reason why we document the properties is to ensure objectivity. We consider it good practice to present the criteria on which we will examine an IMD before we start examining an IMD. Notice that we only describe properties from an information security perspective, properties such as: the patient may not be allergic to the material of the IMD are not considered. In Section 4.4 we summarize the security properties and appoint the place they have on the CIA-triad.

### 3.6.1. Therapy safety

We distinguish four therapy safety properties: Incident Treatment Delivery property, Necessary Treatment Delivery property, Right Treatment Delivery and the Incident Right Treatment Delivery property [65]. First the *Incident Treatment Delivery* property which should ensure that when a treatment is needed the patient receives treatment. Second the *Necessary Treatment Delivery* which ensure that the patient only receives treatment when he is in need of it. The third property is the *Right Treatment Delivery* property which should ensure that when a patient receives treatment he receives the right treatment. The final property is the *Incident Right Treatment Delivery* property which combines property $P_1$ and property $P_2$ to ensure that the patient receives the right treatment when he is in need of it. An active violation of one or multiple of the properties above could result in a serious risk to the patients safety.

[16] `http://gnuradio.org/redmine/projects/gnuradio/wiki`
[17] `http://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion`
[18] `http://code.ettus.com/redmine/ettus/projects/uhd/wiki`

| ID | Property | CIA place |
|---|---|---|
| $p_1$ | Incident Treatment Delivery | Availability |
| $p_2$ | Necessary Treatment Delivery | Integrity |
| $p_3$ | Right Treatment Delivery | Integrity |
| $p_4$ | Incident Right Treatment Delivery | Availability and Integrity |
| $p_5$ | Device Existence Privacy | Confidentiality |
| $p_6$ | Device Type Privacy | Confidentiality |
| $p_7$ | Specific Device Traceability | Confidentiality |
| $p_8$ | Device Data Confidentiality | Confidentiality |
| $p_9$ | Guaranteed Emergency Access | Availability |
| $p_{10}$ | Human Accountable Therapy Modification | Integrity |
| $p_{11}$ | Device Accountable Therapy Modification | Integrity |
| $p_{12}$ | Organisation Accountable Therapy Modification | Integrity |
| $p_{13}$ | Authorized Healthcare Professional Device Traceability | Integrity |
| $p_{14}$ | Authorized Update Source | CIA |
| $p_{15}$ | Authorized IMD update | CIA |
| $p_{16}$ | Authorized Update | CIA |
| $p_{17}$ | Time to Update | CIA |
| $p_{18}$ | Device Data Integrity | Integrity |
| $p_{19}$ | Security Incident Notification | CIA |
| $p_{20}$ | Attack Recognition | CIA |
| $p_{21}$ | Standardized Protocols and Software | Integrity |
| $p_{22}$ | Self Verification | Integrity |
| $p_{23}$ | User Acceptance | CIA |

Table 6: Desired security properties for an IMD

### 3.6.2. Privacy

The IMD could carry a big amount of personal data [56]. For example the name and address of the patient [38] or even more sensitive the medical logbook of the patient [39]. Since this data is medical patient data the device has to comply to several kinds of legislation [56]. In Section 4.4 we will elaborate on the possible impact of a privacy violation. We distinguish four privacy properties: Device Existence Privacy property, Device Type Privacy property, Specific Device Traceability property and Device Data Confidentiality [39]. The *Device Existence Privacy* property states that it should now be possible for an adversary to detect that the there is an IMD active in the area. The *Device Type Privacy* property states that It should not be possible for an adversary to enumerate the device-type when detecting a device signal. Notice that this property is weaker than the device existence privacy property because it allows to show that there exists a device. All medical equipment is required by law to have a Unique Device Identification(UDI)[19]. To satisfy this property, at least the UDI should be hidden. Notice that if the IMD uses a combination of unique communication protocols the observation of these protocols may reveal the device type. For example if $IMD_1$ is the only IMD which uses multiple modulation schemes to communicate during one session an attacker who observes this could learn the device type.
The *Specific Device traceability* property states that when an adversary detects a device and device type it should not be possible to trace it back to an individual. The *Device Data confidentiality.* property states that if an adversary detects an IMD and knows it specifications, it should not be possible for the adversary to extract any personal data from the device.

[19] `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0542:FIN:EN:PDF`

### 3.6.3. Emergency access

When an IMD patient needs immediate surgery, the healthcare professional should be able to turn off the IMD [12, 23, 37]. The healthcare professional needs to be able to turn of the device because some operations are in conflict with the working of the IMD. We define the *Guaranteed Emergency Access* property as the assurance that in case of emergency the healthcare professional will always have emergency access to the IMD. This property does not necessarily have to be in contradiction with an authentication scheme.

### 3.6.4. Accountability

Accountability is the assurance that an identified person takes responsibility and is accountable for a certain action. We split this term in three different definitions. The *Human Accountable Therapy Modification* property assures that every modification in therapy is traceable to an identified healthcare professional which is responsible for that modification. The *Device Accountable Therapy Modification* property assures that every modification in therapy is traceable to an identified medical device which is responsible for that modification. For example by using the UDI. The *Organisation Accountable Therapy Modification* property assures that every modification in therapy is traceable to an identified organisation which is responsible for that modification.

### 3.6.5. Detection and Verification

Because we talk about desired properties we also list some properties which are very hard to satisfy. The *Attack Recognition* property ensures that the device is able to recognise a genuine communication and a non genuine communication. The *Security Incident Notification* property ensures that when a security breach occurs the IMD notifies the patient and the healthcare professional. This notification should go via another channel than the primary channel. The *Standardized Protocols and Software* property ensures that all software, algorithms and protocols used to make the IMD are based on established standards. The *Self Verification* property states that every patient should be able to verify (if he is knowledgeable enough) that his or her IMD satisfies the documented security properties. This property is a variation of the Kerckhoffs [54] Principle that states that: "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge".

### 3.6.6. Patching, updating and incident response

The *Authorized Healthcare Professional Device Traceability* property ensures that an authorized and responsible healthcare professional is able to trace a patient or patients under his supervision based on an IMD type or UDI. This property is necessary to mitigated vulnerabilities for a specific device or device type. The *Authorized Update Source* property ensures that an IMD could only be updated from an authorized source. The *Authorized IMD Update* property ensures that an IMD could only update an authorized IMD update. The *Authorized Update* property combines the previous two properties and ensures that an IMD could only update an authorized IMD update via an authorized source. The *Time to Update* property ensures that every IMD for which a specific security vulnerability is found is updated within an acceptable time.

### Other

The *Device Data Integrity* property states that all data on the IMD may only be modified by an authorized, authenticated healthcare professional. The *User Acceptance* [71] property states that every security solution should be accepted by the big majority of the IMD patients.

# 4. IMDs and their security

We found that current IMD risk assessments methodologies focus more on safety than on information security [67, 13]. These methods focus on, for example: the risk of surgery [36] or the risk of complications within the human body. We did not find any IMD risk assessment that involves threats and vulnerabilities related to IMD hacking or unauthorized access [67] to the IMD. In this section we do a risk assessment on our abstract IMD and we specifically focus on information security. This risk assessment methodology follows some building blocks provided by ISO/IEC 27001:2009 [14] on which we elaborate in Section 5. In this section we start by analysing the possible assets of our abstract IMD. Then we enumerate attack scenarios and vulnerabilities. Finally we list the threats, possible attackers and try to determine the impact of successful exploitation.

## 4.1. Asset analysis for IMDs

Depending on who is making the analysis, the assets could differ. We do not incorporate business assets such as corporate reputation or liability. With this in mind we distinguish three assets:

1. The IMD as hardware.
2. The patient health which depends on the IMD.
3. Confidential data stored in the IMD.

## 4.2. Attacks & attack scenarios for IMDs

There are many vulnerabilities which are commonly found in information systems. Those vulnerabilities are widely discussed on websites such as: Bugtraq[20] or CVE[21]. In this section we list attack scenarios which may be applicable to an IMD attack.

**Battery draining.** Battery draining is an attack that exploits the limited battery capacity of the IMD [44, 80, 72]. There are multiple ways to attack the battery. An option is to repeatedly generate communication or authentication requests to an IMD [44]. Even if the communication or authentication requests fails, the IMD is still wasting battery power. A second option is to keep a communication channel open as long as possible. Both attacks result in a continuously active IMD draining its battery power.
**Communication jamming.** Communication jamming is an attack that focuses on the wireless communication between the IMD and external hardware.

[20] http://www.securityfocus.com/
[21] http://cve.mitre.org/

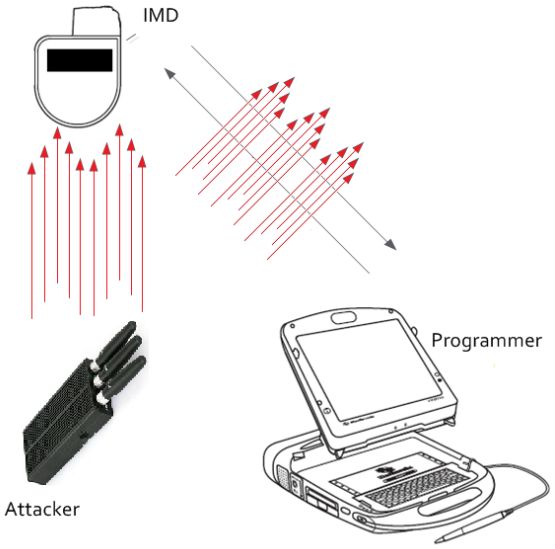



Figure 19: Result passive eavesdropping between pacemaker and pacemaker programmer [38]

A method for communication jamming is to keep sending powerful enough signals on the same frequency used by the IMD [9]. This causes noise and makes it impossible for the IMD to construct the original message. This causes unavailability of communication but does not necessarily effect the availability or working of the IMD itself.

**Passive eavesdropping.** This attack focuses on an IMD that is communicating with an external device. The attacker intercepts the communication between the IMD and the external device.

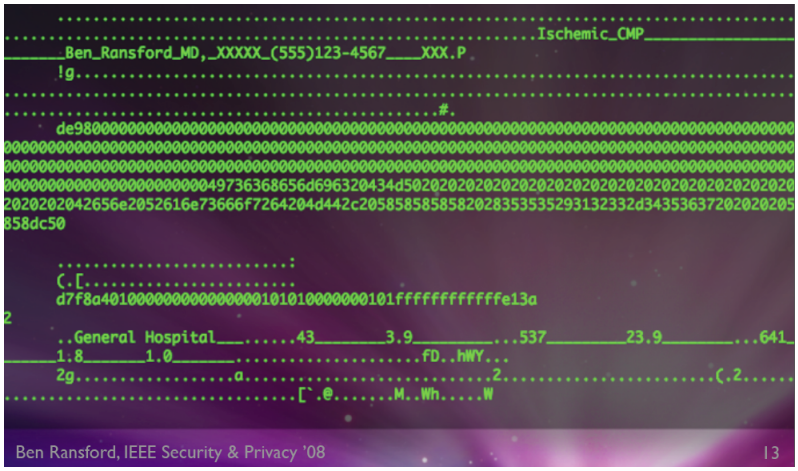



Figure 20: Result passive eavesdropping between pacemaker and pacemaker programmer [38]

If the communication between the IMD and the external device is unencrypted an attacker can understand the content of the communication. For a pacemaker, this case is demonstrated by Halperin et al. [38]. As shown in Figure 20 the researchers where able to read: the implanting physician, the diagnosis and the hospital. The researchers where also able to read: the device state, Patient name, Patient date of birth, pacemaker model and more [38]. If the communication is encrypted the attacker could store the communication trace for offline cryptographic attacks.

**Man in The Middle attack (active eavesdropping).** This attack focuses on an IMD that is communicating with an external device. The difference between "active eavesdropping" and

"passive eavesdropping" is that the attacker could tamper with the transmission in the active case while, in passive case the attacker only listens to the transmission [85]. The basic concept of the "Man in The Middle attack" begins with the interception of the communication between the IMD and the external device [73]. Then, the attacker replies the communication to its original destination while impersonating the original source. Through this procedure the attackers is able

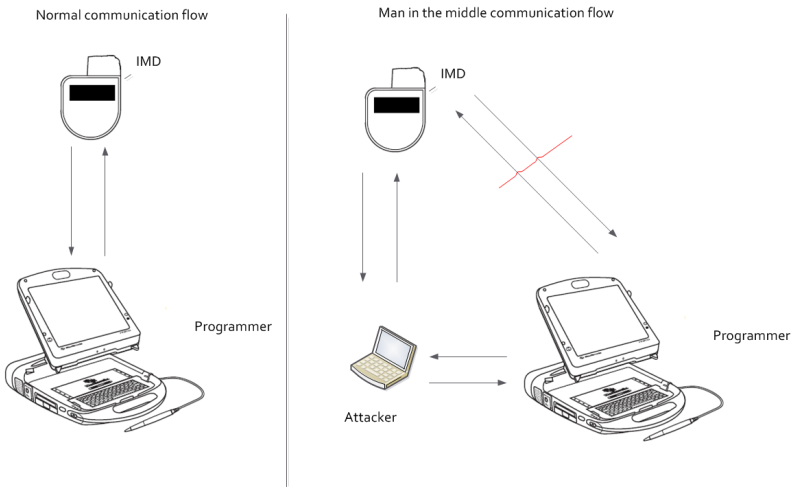



Figure 21: IMD Man in the middle attack

to eavesdrop and manipulate the transmission. If the communication is encrypted the attacker could try to exploit communication protocol weaknesses.

**Weak or non-existing authentication.** This attack focuses on the authentication of the IMD. If the IMD allows access from external devices it should ensure that an appropriate authentication protocol is used. If there is no authentication, every attacker could use or abuse the functionality of the device. Weak authentication forms range from bad passwords [8] to cryptographic weaknesses [85] and protocol weaknesses [74].

**Software / firmware vulnerabilities.** Software / firmware vulnerabilities of an IMD could be exploited by an attacker who is able to communicate with an IMD but does not have access to the software / firmware. There are multiple taxonomies and attack scenarios for software vulnerabilities [87]. We deem the lack of lack of input validation, security feature implementation faults, time and state attacks, unsafe failure and environment vulnerabilities applicable to IMDs. The lack of input validation is also a commonly found vulnerability in software. For example a buffer overflow exploit could result in unauthorised code execution or crashing the software [8]. Security feature implementation faults are faults related to the logic of the code. They vary from wrong implementation of random generators to not dropping root privileges when performing a privileged operation. If an IMD has different levels of privileges, they can sometimes be bypassed by privilege escalation which may lead to information leakage. If the software does not fail appropriately, it may try to reset itself with more privileges than it had before. If the software provides detailed error information it may unintentionally leak confidential information. For example if a stack trace is displayed including the content of various strings with information about the patient this discloses confidential information. Environment vulnerabilities are vulnerabilities caused by the software / firmware development software. Debug functionality could lead to higher privileges and test functions may provide a way to bypass authentication or exhaust the battery of the device. During the security assessment the security assessor should pay attention to these vulnerabilities.

**Electromagnetic interference.** Electromagnetic interference could affect the working of the IMD. For example, the pacemaker could stop delivering the stimulating pulses or cause the pacemaker to ignore the heart's own rhythm and deliver pulses at a fixed rate [58, 92]. This affect could for example be triggered by: metal detectors, cell phones or a Magnetic resonance imaging (MRI) scan [58, 67].

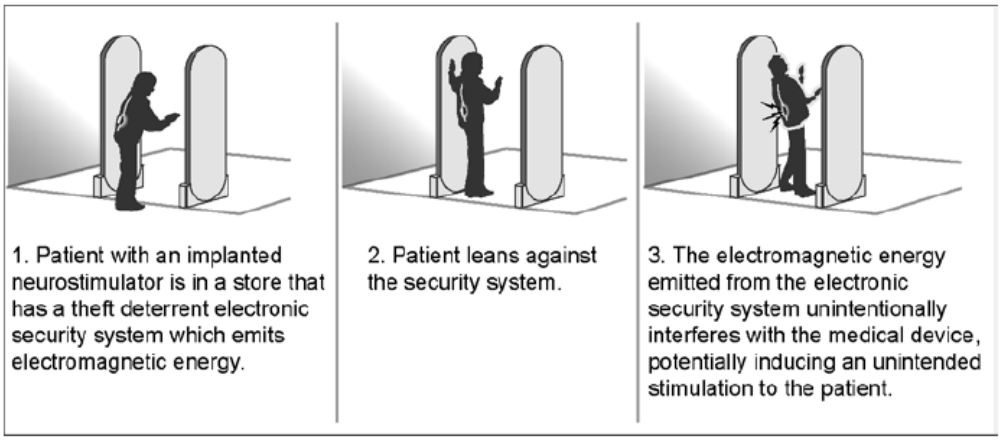



Figure 22: Example of electromagnetic interference affecting IMD [67]

As shown in Figure 22, the exposure of an IMD to an electromagnetic field may result in unexpected behaviour.

**Radio traffic analysis** Traffic analysis is a method which tries to extract information by analysing communication patters [46]. For example (as shown in Figure 23) observing communication between 402 MHz and 405 MHz [19] leaks the information that in a small area someone is using an IMD.

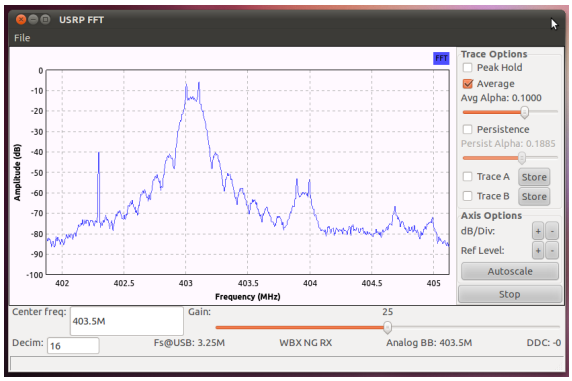



Figure 23: Observation of an IMD communication trace

Characteristics of the radio communication may leak more information [46] about the device type and its configuration.

**Social engineering.** Social engineering from an information security perspective is: "the art of hacking a system by manipulating the people who operate that system" [8]. The attack tries to exploit the vulnerability of untrained or unaware healthcare employees or patients. The scenarios for a successful attack on an IMD via social engineering are only limited by the imagination of the attacker [64]. An example of a social engineering attack could be an attacker calling the patient pretending to be the healthcare professional and asking the patient to adjust the IMD settings.

**Backdoor / trojan horse.** A backdoor or trojan horse is a piece of malicious software that grants the owner of it unauthorized access to the device. The backdoor could be placed by an authorized attacker or an unauthorized attacker which obtained access by exploiting another vulnerability. In some cases the backdoors are placed with good intentions for example as a substitute for a decent emergency protocol [39].

## 4.3. Threat and attackers analysis for IMDs

In this section we try to enumerate possible threats and attackers with their attack strategies. An information security threat is an event or danger that might adversely affect the IMD. In Table 7 we distinguish five different threats for IMDs.

| ID | Threat | Description |
|---|---|---|
| $T_1$ | Unauthorized access to the IMD. | Allows an attacker to control the device. |
| $T_2$ | Data leakage from the IMD. | Personal information of the patients leaks from the device to the public. |
| $T_3$ | Malfunctioning / unexpected behaviour of the IMD. | The device its behaviour becomes unpredictable. It could send random signals or random change data / therapy settings. |
| $T_4$ | Harmful behaviour of the IMD to the patients health. | The device its behaviour results in an event which is harmful to the patient. |
| $T_5$ | Denial of service of the IMD. | The device stops working or does not perform properly. |

Table 7: Potential threats to IMDs.

We also distinguish five different attackers.
We define the **non-skilled attacker** as an attacker with no background in information security. An objective for this attacker could be to harm the patient without letting other people find out. To accomplish this, the attacker could modify external hardware or change the medication. This attacker has limited resources and cannot afford specialized equipment.
We define the **skilled attacker(s)** as an attacker with knowledge of how to perform a security attack. The attacker could intercept and modify transmissions but has limited resources. The attacker knows of the existence of professional equipment but cannot afford it. The attacker is therefore limited in his resources to inexpensive home-made equipment.
We define the **inside attacker(s)** as an attacker who has access to equipment for reprogramming an IMD. The insider is aware of the device architecture and its weaknesses.
We define the **well funded organisation** as an organisation who could hire or bribe the expertise from both inside and outside experts.
We define the **nation state** as an organisation which could do the same as the well funded organisation but with almost unlimited resources.
Due to the possible unpredictable effect of electromagnetic interference on an IMD, **nature** could be considered a danger for an IMD but not an attacker. This threat could randomly interfere with radio communication or activate an IMD when an magnetic trigger is used for activation.
An **accident** initiated by the patient or healthcare professional could be a threat to the patient. For example a healthcare professional could misconfigure an IMD. However, we consider an accident as a derivative threat which is encapsulated by the non-skilled attacker and the inside attacker. Therefore we do not consider this as an separate case.
Finally the device could be attacked by **malware**. Malware is a container definition for a number of types of malicious code [8]. Examples of malware are: viruses, logic bombs, trojan horses, rootkits or keyloggers [83]. As of today, we are not aware of the existence of malware specifically targeting IMDs. There exist however malware that attacks specific medical equipment like AEDs [40]. Because Malware is an derivative attacker produced by one of the threats mentioned above we do not consider this as an isolated case. Notice that the skill level for each succeeding is higher than its predecessor except for the final one: nature.

## 4.4. Likelihood determination

The likelihood of successful exploiting a vulnerability could depend on an infinite amount of parameters. Examples of these parameters could be: current time, current location, neighbourhood

or reputation. Therefore we deem it not possible to make a statement about the likelihood for a vulnerability. Based on the five type of attackers: Non skilled attacker ($A_1$), Skilled attacker ($A_2$), Inside attacker ($A_3$), Funded organisation ($A_4$) and Nation states ($A_5$) we could determine if an attacker may be capable of performing a certain attack.

| **Vulnerability type** | **Attackers** |
|---|---|
| Battery draining | $A_2$,$A_3$,$A_4$,$A_5$ |
| Communication jamming | $A_2$,$A_4$,$A_5$ |
| Eavesdropping | $A_2$,$A_4$,$A_5$ |
| Man in The Middle attack | $A_2$,$A_4$,$A_5$ |
| Weak or non-existing authentication | $A_2$,$A_4$,$A_5$ |
| Software / firmware | $A_2$,$A_4$,$A_5$ |
| Electromagnetic interference | $A_3$,$A_4$,$A_5$, nature |
| Traffic analysis | $A_2$,$A_4$,$A_5$ |
| Social engineering | $A_1$,$A_2$,$A_3$,$A_4$,$A_5$ |
| Backdoor / trojan horse | $A_3$,$A_4$,$A_5$ |

Table 8: Vulnerabilities and attackers capable of exploiting.

#### Impact of successful exploitation

In this section we evaluate the impact of successful exploitation of a vulnerability. It is important to notice that we cannot evaluate the impact on the abstract model of the IMD since each IMD could have its own unique impact. In Section 6 we give an example of the impacts based on a real IMD.

The most disastrous effect of successful exploiting a vulnerability by a threat could be the death of one or more patients. In this worst case scenario a signal is send which modifies the working of the IMD to a deadly therapy. The same result could be established by a firmware update of the IMD which could reprogram the IMD to a lethal therapy. Both 'black swans' are only possible if the specific IMD is capable of killing a human. However, there are insulin pumps which carry enough insulin to kill an individual and have the option to deliver a bolus dose.

Another impact of successful exploitation could be a privacy breach. Most IMDs contain a huge amount of personal data. If an attacker could eavesdrop on the communication or obtain (unauthorized) access to the IMD there could be a privacy breach. Privacy of citizens is under ongoing debate. For our analysis we assume the worst case result of a privacy violation: people may lose their jobs, pay a higher insurance fee [12], miss a promotion, get discriminated [12] or get embarrassed by the fact that they need an IMD for their health.

The third possible impact may be unavailability of the device. This may be caused by a battery exhaustion attack. Needless to say, this could be a risks for the patient safety because the replacement of an IMD battery implies surgery and surgery comes with the risk of infection or death [36]. The unavailability may also focus on the communication. In that case, the communication may be jammed which could delay fine tuning of the therapy. This does not necessarily impact the patient health directly.

We rated the impact of the violation of each property from Section 4.4 in Section 4.4.

| **PID** | **Property** | **Impact violation classification** |
|---|---|---|
| $p_1$ | Incident Treatment Delivery | Lethal |
| $p_2$ | Necessary Treatment Delivery | Lethal |
| $p_3$ | Right Treatment Delivery | Lethal |
| $p_4$ | Incident Right Treatment Delivery | Lethal |
| $p_5$ | Device existence privacy | Low |
| $p_6$ | Device-type privacy | Low |
| $p_7$ | Specific-device traceability | Medium |
| $p_8$ | Device data confidentiality | High |
| $p_9$ | Guaranteed emergency access | Lethal |
| $p_{10}$ | Human Accountable Therapy Modification | Low |
| $p_{11}$ | Device Accountable Therapy Modification | Low |
| $p_{12}$ | Organisation Accountable Therapy Modification | Low |
| $p_{13}$ | Authorized Healthcare Professional Device Traceability | Medium |
| $p_{14}$ | Authorized update source | Medium |
| $p_{15}$ | Authorized IMD update | Medium |
| $p_{16}$ | Authorized update | High |
| $p_{17}$ | Time to update | Medium |
| $p_{18}$ | Device data integrity | High |
| $p_{19}$ | Security breach notification | Medium |
| $p_{20}$ | Attack recognition | Medium |
| $p_{21}$ | Standardized protocols and software | Medium |
| $p_{22}$ | Self verification | Low |
| $p_{23}$ | User acceptance | Low |

Table 9: Impact on property violation

# 5. Standards for security assessments

In this section we discuss two information security standards.

1. OWASP ASVS
2. SANS 20 Critical Security Controls (CSC) v4.0

We choose to evaluate these standards for their applicability because they are widely used [93], open and free available under the Creative Commons Unported License. This means that we did not need to purchase the standards and we are free to quote them. As we have seen in Section 3 SCADA systems share more than half of the characteristics we defined with IMD characteristics. On a more abstract level we found that the connection between the HMI and MTU is comparable to the connection between the IMD and the IMD programmer. We also found that the historian at the "controller system" level is comparable to the data storage inside the IMD. Because the 20 CSC list is used to improve the security of SCADA systems [41] and SCADA systems bear similarities with IMDs on abstract level and characteristic level, we choose to evaluate the 20 CSC for their applicability to an IMD security assessment. As we have also seen in Section 3 (Web) information system characteristics are around one third similar to IMD system characteristics. Although the similarities in characteristics are less than half applicable, we found that the similarities in the abstract architecture are stronger. The relation between the client and the web interface can be compared to the relation between the IMD programmer and the IMD. Also the data storage of a (Web) information system is comparable to the data storage inside the IMD (although the IMD data storage is far more limited). The similarity is that both systems are capable of reading, writing and modifying data [38, 34, 71, 43, 76]. Because Web applications have become prevalent around the world [63] and the OWASP ASVS is used to improve their security [93, 18], we choose to evaluate the OWASP ASVS for its applicability to an IMD security assessment. Because we want to propose a security assessment methodology in Section 7 and we want to use a part of ISO/IEC 27001:2009 in this security assessment methodology we also discuss a part of this security management standard. We rate the applicability of the verification and security controls based on the characteristics and desired security properties we enumerated from the abstract IMD in Section 3.

## 5.1. The Open Web Application Security Project

The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organization focused on improving the security of software[22]. In 2009 the OWASP released the Application Security verification standard (ASVS). The ASVS provides four levels of verification: automated, manual, design and internal verification for which each level is an extension of the previous level. The ASVS defines 14 security requirements areas, having each its own verification criteria.

### Security Architecture Documentation Requirements

This set of verification requirements focuses on documentation. The verification of these requirements contribute to the verification of all desired security properties. IMDs should be properly documented because documentation could provides important information for a security assessment. The security assessor should assure that all IMD components (1.1, 1.2, 1.4 and 1.5) and the IMD architecture (1.3) is documented. Finally the assessor should verify that a threat model (1.6) is available.
Since all verification requirements are applicable we deem this section applicable to an IMD security assessment.

[22] `https://www.owasp.org/index.php/Main_Page`

#### Authentication Verification Requirements

This set of verification requirements focuses on the authentication of the software product. The verification of these requirements contribute to the verification of desired security properties: $P_6$, $P_7$, $P_8$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{14}$, $P_{15}$, $P_{16}$ and $P_{18}$. IMDs should only have three actors: The patient, The healthcare professional and the IMD developer. The healthcare professional should be in charge of authentication management. This should ensure that the protocol for password management is followed and that the IMD-user cannot lose his or her password (2.9) permanently. For safety reasons it should not be possible for an IMD to lock a user account (2.3) or to suspend the login procedure (2.2). Verification requirements which focus on these options are therefore not applicable. In addition, special care should be taken in evaluating the emergency protocol. The authentication should take place on the device. Therefore, a centralised or distributed authentication mechanism (2.5) should not be applicable.
We deem this section more than half applicable to an IMD security assessment.

#### Session Management Verification Requirements

This set of verification requirements focuses on the session management verification requirements. The verification of these requirements contribute to the verification of desired security properties: $P_6$, $P_7$, $P_8$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{14}$, $P_{15}$, $P_{16}$ and $P_{18}$. IMDs are implanted into the body therefore it is a risk full and expensive operation to change a battery. Therefore, the architecture of an IMD is designed to use limited battery power, for instance, it does do not have an internal clock. Therefore, most IMDs do not use sessions and are stateless, so that session time outs (3.4, 3.3) are not applicable. IMDs do use a protocol for each command or transmission that could have some session like properties. For stateless protocols there are standard frameworks to realize something similar to session management (3.1). Also principles that indicate wrong or good implementation of cryptographic measurements (3.6, 3.7, 3.8, 3.10, 3.11) are applicable to IMDs. Since IMDs do not distinguish pages or functions and does not have the concept of users or privileges, authentication may seem pointless. However, as we will describe in Section 8.2 it may be possible that IMDs will adapt a concept of privileges in the future. Therefore verifying that functions that require authentication enforce authentication (3.5) may be applicable in the future. We deem this section more than half applicable to an IMD security assessment.

#### Access Control Verification Requirements

This set of verification requirements focuses on the access control requirements. The verification of these requirements contribute to the verification of desired security properties: $P_6$, $P_7$, $P_8$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{14}$, $P_{15}$, $P_{16}$ and $P_{18}$. IMDs do not distinguish pages or functions and does not have the concept of users or privileges, so that access control may seem pointless. However, as we will describe in Section 8.2 it may be possible that IMDs will adapt a concept of privileges in the future. With this concept the protection of: functions (4.1), data (4.3) and services (4.6) becomes applicable. To enforce this, proper implementation is of authentication (4.7), tamper safe policies (4.10), input validation (4.13) and secure failure (4.8) is necessary. Secure failure must be extended to safe and secure failure. As will will suggest in Section 8.3, accountability (4.14) may be implemented for IMDs. Some of the access control verification requirements are very specific for web applications. The protection of URLs (4.2), web directories (4.5), presentation layer (4.9), central access control (4.10) the server (4.11) is more applicable on these systems than on IMDs. We deem this section more than half applicable to an IMD security assessment.

#### Input Validation Verification Requirements

This set of verification requirements focuses on input validation. The verification of these requirements contribute to the verification of desired security properties: $P_3$, $P_4$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{14}$, $P_{15}$, $P_{16}$, $P_{18}$ and $P_{21}$. All verification requirements are applicable when the word “server” is replaced by “IMD”. We deem this section applicable to an IMD security assessment.

#### Output Encoding/Escaping Verification Requirements

This set of verification requirements focuses on output encoding/escaping of IMDs. The verification of these requirements contribute to the verification of desired security properties: $P_3$, $P_4$, $P_{20}$ and $P_{21}$. An IMD is implanted in the human body. Therefore, no visible output can be displayed. Every display is on a secondary device that threats the IMD data as input. As we will see, some IMDs may use a strict format for data storage. Therefore, it is good practice to specify and enforce an output format based on the allowed characters and length (6.2, 6.3 and 6.8). If this is implemented, one should verify that a single security control takes care of the output formatting (6.9). With these of output validation, rules that focus on specific web systems like html encoding (6.1), SQL escaping (6.4), XML escaping (6.5) LDAP escaping (6.7) and system command escaping (6.8) are not applicable. We deem this section half applicable to an IMD security assessment.

#### Cryptography Verification Requirements

This set of verification requirements focuses on the cryptographic modules. The verification of these requirements contribute to the verification of desired security properties: $P_6$, $P_7$, $P_8$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{14}$, $P_{15}$, $P_{16}$, $P_{18}$, $P_{21}$ and $P_{22}$. IMDs are implanted into the human body which makes battery replacement a risk full and expensive operation [36]. To limit battery replacement as much as possible, most IMDs are programmed to be as energy efficient as possible [38, 34, 43, 69, 71] and therefore do not use cryptography [38, 62]. If Cryptography is used every verification requirement would be applicable. But because it is not used we deem non of the requirements applicable for this moment. We deem this section not applicable to an IMD security assessment.

#### Error Handling and Logging Verification Requirements

This set of verification requirements focuses on logging and error handling. The verification of these requirements contribute to the verification of desired security properties: $P_{10}$, $P_{11}$, $P_{12}$, $P_{19}$ and $P_{20}$. If an error occurs on the IMD it might have effected the therapy. Therefore it is essential that the healthcare professional knows what has happened. This knowledge could be provided by logging. Since IMDs do not have a clock it is not possible to include a time-stamp in the logs (8.6.1). Since IMDs (currently) don't have an IP address it is not possible to include the IP address in the logs (8.6.5) either. However, every device that connects with the IMD might have an unique hardware ID. If so, this should be included in the logs. Further are all verification requirements in this section applicable when the word "server" is replaced for "IMD". We deem this section more than half applicable to an IMD security assessment.

#### Data Protection Verification Requirements

This set of verification requirements focuses on data protection. The verification of these requirements contribute to the verification of desired security properties: $P_6$, $P_7$, $P_8$ and $P_{18}$. Verification requirements that focus on the cache (9.1) or on HTTP (9.3) are not applicable since the IMD does not use these techniques. As we have described in Section 3 the connection of the IMD may reveal confidential information. We therefore think that this set of verification requirements should be extended with three additional requirements:

1. Verify that the method of connecting does not reveal that a medical device exists.
2. Verify that the method of connecting does not reveal the type of the device.
3. Verify that the method of connecting does not reveal the location or the device owner.

We deem this section less than half applicable to an IMD security assessment.

#### Communication Security Verification Requirements

This set of verification requirements focuses on the communication. The verification of these requirements does not contribute to the verification of desired security properties. Since IMDs do not use cryptography to save battery power [38, 62] it cannot build a TLS (10.1,10.2, 10.3, 10.4 and 10.8) connection or support a PKI structure (10.1 and 10.5). The authentication of the connection (10.6) and access control of the connection (10.7) requirements are not applicable as described in the previous paragraphs. The character encoding of the connection is, as discussed in the input/output verifications paragraphs applicable to IMDs. We deem this section less than half applicable to an IMD security assessment.

#### HTTP Security Verification Verification Requirements

We deem this section not applicable to an IMD security assessment because IMDs do not use HTTP.

#### Security Configuration Verification Requirements

This set of verification requirements focuses on configuration. The verification of these requirements contribute to the verification of desired security properties: $P_1$, $P_2$, $P_3$, $P_4$, $P_5$, $P_6$, $P_7$, $P_8$, $P_{10}$, $P_{11}$, $P_{12}$, $P_{13}$, $P_{14}$, $P_{15}$, $P_{16}$, $P_{17}$ and $P_{18}$. IMD configuration files cannot be stored in a authorized location (12.1, 12.2). Currently IMDs do not have access control or the concept of authentication [32]. Without authentication or access control it is not possible to have an authorized location. However, we could verify if an IMD logs changes to the configuration (12.3). We could also verify if the configuration is in human readable format (12.4). We deem this section less than half applicable to an IMD security assessment.

#### Malicious Code Search Verification Requirements

This verification requirement focuses on malicious code. The verification of these requirements contribute to the verification of all desired security properties. All code for the IMD needs to be checked for malicious code built in during development or installation time (13.1), therefore all verification requirements in this set are applicable. In addition, it is important to check if all debug and unit test modules are disabled or removed after installation (13.2). We deem this section applicable for an IMD security assessment.

#### Internal Security Verification Requirements

This set of verification requirements focuses on the internal security requirements. The verification of these requirements contribute to the verification of all desired security properties. IMDs are not suitable for multiple applications (14.1 and 14.3) therefore verification requirements that focus on multi application security on one server / device are not applicable. We could verify if security controls are simple enough so that developers will use them (14.2) Therefore all verification requirements in this section are applicable to IMDs. We deem this section less than half applicable to an IMD security assessment.

#### Applicability

We evaluated all verification requirements. If a topic was missing verification requirements which where specific for IMDs we added them to that topic. If a complete section was missing (for example about the limited battery capacity) we did not add it (yet). In total we evaluated 123 security verification requirements on their applicability on IMDs. We found that more than half of the verification requirements are applicable to an IMD security assessment. Therefore, we deem this security verification standard more than half applicable to an IMD security assessment.

## 5.2. SANS

The SysAdmin, Audit, Network, Security (SANS) Institute provides a list of 20 security controls[23]. The SANS controls focus on a whole network and therefore many of them are not applicable to IMDs. Because the applicable controls are more abstract than the OWASP verification requirements it makes no sense to relate the controls to the desired security properties in this case.
The first control:"Inventory of Authorized and Unauthorized Devices" is applicable when the IMD is connecting to multiple remote devices or if multiple remote devices contain IMD data. The latest scenario: remote devices containing IMD data, is applicable because there are vendors which have iphone apps to view the patient data[24].

The second control: "Inventory of Authorized and Unauthorized Software" is not applicable since the device should not run user installed software and only accept firmware updates enforced by the IMD-programmer.

The third control: "Secure Configurations for Hardware and Software on Laptops, Workstations, and Servers" is applicable to the IMD security. As we will see in Section 6, the IMD communicates with these object via a programmer device. Therefore it is necessary to ensure they have a secure configuration as well.

The fourth control: "Continuous Vulnerability Assessment and Remediation" is applicable, it is good practice to make vulnerability assessments on critical systems. An IMD is a critical system. However, as discussed in Section 3, remediation of vulnerabilities is hard.

The fifth control: "Malware Defenses" is not applicable to the IMD. This control is more applicable to the development process and the secondary hardware which connect to the IMD.

The sixth control: "Application Software Security" is applicable. It is essential to prevent successful (software ) attacks on the IMD. Even when considered that the IMD has only one application, this one should be secure.

The seventh control: "Wireless Device control" is half applicable. This control focuses on wireless access points. The access point is part of the environment and not of the IMD. Therefore it is not directly applicable. However, several IMDs communicate via a wireless channel. Therefore we should extend the control with the privacy properties from Section 3 and evaluate them.

The eighth control: "Data recovery capability" is half applicable to the IMD. Due to the memory limitation of the IMD, the IMD should not backup within itself. However, ensuring that a backup of the IMD is made regularly and stored in a proper way should be part of an IMD security assessment.

The ninth control: "Security Skills Assessment and Training to Fill Gaps" is applicable. Developers, healthcare professionals and patients could all make mistakes or be targeted by a social engineering attack. Therefore anyone who has something to do with an IMD should have had a proper training. The training of the patient could be given during the introduction to the IMD (when it is just implanted). Besides the functionality the patient should be educated about social engineering. Hospitals could adopt the banking policy: "we do not call or e-mail you". With this policy patients could be trained to not answer phone calls or e-mail asking for their IMD data. Developer should follow secure coding and security trainings. Finally healthcare professionals should be trained to be able to recognise possible threats on IMDs.

The tenth control: "Secure configurations for Network devices such as Firewalls, Routers, and

[23] `http://www.sans.org/critical-security-controls/`
[24] ⋆⋆⋆⋆⋆

switches" is not applicable to the IMD. However, since the IMD data is often stored on a regular (hospital) network, this control is applicable to that network.

The eleventh control: "Limitation and Control of Network Ports, Protocols, and Services" is applicable. For software security it is important that the device only provides functionality which is necessary to operate. Nice to have features which don't improve the patients health or the device its security and should therefore not be implemented.

The twelfth control: "Controlled use of administrative Privileges" is currently not applicable. IMDs do not use authentication and therefore do not have the concept of privileges [32]. However as we will describe in Section 8.2 it may be possible that IMDs will adopt a concept of privileges in the future.

The thirteenth control: "Boundary Defense" is applicable to IMDs. It could be applied to the physical layer (acceptance of signals), to the software layer and to the procedural level. One could for example segregate the battery, one part for communication and one part for critical operations. We will elaborate on this solution in Section 8.

The fourteenth control: "Maintenance and Analysis of Security Audit Logs" is half applicable. It is not possible to conduct maintenance operations on an IMD. These operations are not possible because IMDs are inside the human body and the risks of surgery are almost always bigger than the risk of no maintenance. It is only possible to perform a minimal form of logging on an IMD. IMDs have limited storage space and therefore limited logging capacity. However, when the healthcare professional and the patient have a follow up session, the log files from the previous period could be downloaded. On these log files it is possible to perform a security audit. However, this security audit would be very limited. Since IMDs do not have the concept of authentication [32] or time one cannot audit for unauthorized modifications. However, based on the electronic health records a healthcare professional could verify if the therapy is still correct.

The fifteenth control: "Controlled Access Based On Need to Know" is not applicable. Both the healthcare professionals and the patient should have access to all the data. The patient because it is his data and the healthcare professional for his professional judgement. Other users should not have access to the data from the device. If it is necessary for third parties to process the device data it should be provided by the previous described users and not directly through the IMD.

The sixteenth control: "Account monitoring and control" is not applicable, since IMDs do not have accounts, users or authentication [32]. Even if IMDs would have user accounts it would not be a good option to actively monitor them because of the limited battery capacity.

The seventeenth control: "Data loss prevention" is not applicable to an IMD. Because of the limited battery capacity a high backup frequency or other power consuming data loss prevention mechanisms are not an option for IMDs.

The eighteenth control: "Incident Response Capability" is half applicable. This control does not directly reflect on the IMD. The IMD vendor however should have a clear policy on this topic. During an IMD security assessment one should verify that the IMD vendor has this policy.

The nineteenth control: "Secure network engineering" is not applicable to the hardware itself. However if the device operates in a networked environment it is applicable to that network.

The twentieth control: "Penetration Tests and Red Team Exercises" is half applicable to IMDs. Red Team exercises are not possible on operating IMDs. We find it unacceptable to test on an implanted IMD because of the risk to the patients life. Therefore only penetration testing on

explanted devices is possible.

#### Applicability

The sans controls are more abstract than the OWASP verification requirements. We found seven controls applicable, five controls half applicable and eight controls not applicable to an IMD. Therefore we think the SANS CSC 20 is around half applicable to an IMD security assessment.

## 5.3. ISO/IEC 27001:2009

The "International Organisation for Standardization" (ISO) is an international organisation with 164 member countries which try to develop international standards[25]. The ISO/IEC 27001:2009 defines the fundamental principles, concepts and vocabulary for the information security management system (ISMS) [15]. We want to use the risk assessment process (4.2.1) off the standard. We do not need the organisational security parts from ISO/IEC 27001:2009 nor the clauses about: management responsibility, management commitment, resource management, management review, certification. The ISO/IEC 27001:2009 risk assessment process (4.2.1) [15] is defined by six steps:

1. Identify the assets.
2. Identify the threats to the confidentiality, availability and integrity of those assets.
3. Identify the vulnerabilities .
4. Access the possible impact of those threats.
5. Access the likelihood of those events occurring.
6. Evaluate the risk.

Almost all steps are applicable for an IMD security assessment.

[25] http://www.iso.org/iso/home/about.htm

# 6. Security assessment on a CRT

In this section we perform a security assessment on a real IMD. By assessing the IMD we try to achieve three goals. First, we want to investigate how many of our desired security properties from Section 3 are satisfied by this IMD. Second, we want to investigate if the threats and attacks from Section 4 are applicable to this IMD. Third we want to use the experience from this assessment to propose a security assessment methodology.

## The device

We choose to evaluate the ⋆⋆⋆⋆⋆. The⋆⋆⋆⋆⋆ is a Cardiac Resynchronization Therapy (CRT) heart device. To improve the patient health the CRT device stimulates the two ventricles of the


Figure 24: ⋆⋆⋆⋆⋆

heart to beat at the same time by delivering tiny electrical pulses to both sides. Besides sending electrical impulses for therapy purposes the device is also able to collect data and transmit the data to external sources.

### Assets

We identified three assets for the CRT device:

1. The CRT device as hardware.
2. The patient health that should be protected by the CRT device.
3. The patient data captured and stored by the CRT device.

### Stakeholders

The stakeholders for our research are: the hospital who gave us the opportunity to test on a pacemaker programmer, the Technical University of Eindhoven (TU/e), Deloitte, society as a whole because it could benefit from increased IMD security and we, as researchers.

## 6.1. Obtaining a device

For accurate results it was necessary to perform a security assessment on a real IMD. To obtain an IMD we contacted three big IMD vendors. Unfortunately none was willing or able to cooperate with us at that time. Because e-mail conversation are private we will not publish the names of the companies or their representatives. However, three quotes from the replies are worth mentioning for our security assessment because those responses give an insight in how some vendor representatives think about IMD security. For the sake of simplicity we translated all the statements to English.

> "While the likelihood of a criminal security breach of a medical device is low, industry is addressing device security in the design development process in order to safeguard patient safety."

This statement seems not to be substantiated with numbers or formal research. Although a criminal security breach seems unlikely because of the absence of a simple criminal business model, the risk of a "normal" security breach remains. Besides, medical malware has already been found and reported [40] which could also threaten the security of a medical device.

> "The theoretical case that an individual with specialised equipment and considerable prior knowledge could shut down a pacemaker, is comparable to the case that someone steals commercial equipment from the hospital or manufacturer to shut down the pacemaker."

In this statement, the attack is called a theoretical attack. However, researchers have demonstrated the attack is performable in practice [38].

> "In both cases one has to be really close to the pacemaker to reprogram it. It is nothing more then a research for sensation to achieve a nice headline which unfortunately brings unnecessary agitation on to the pacemaker patients."

We disagree with the part of the statement that suggests that the research is only for sensational purposes. As shown in Section 1 we expect an increasing amount of European citizens depending on implantable medical devices to stay alive [28] or to improve the quality of their life. Because IMDs introduce the threat of an unauthorized hacker harming a patient via his or her IMD [62, 38, 42, 37, 23], we think that researching IMD security is necessary.

## 6.2. Precautions

### Ethical barriers

As ethical barrier we follow five principles.

1. We do not test on real patients.
2. We do not test on animals.
3. We will not try to exploit an in use application.
4. We will not disclose a complete and working description of an attack that could endanger someone's life.
5. We will not disclose real patient data.

### 6.2.1. Responsible disclosure

Because of the possible social unrest and possible health risks for the patients when an IMD vulnerability is publicly published, we decided to write a responsible disclosure procedure. We consider our security assessment confidential and only share it with the participating hospitals, researchers from the TU/e and colleagues from Deloitte. We will only publicly report vulnerabilities after contacting the manufacturer. In the mean time we decided to publish a public version of this thesis. This version differs from the confidential version in the sense that we replaced the name of the vendor(s) with stars (⋆⋆⋆⋆⋆), hide vendor revealing images with a confidential image, hide texts that can reveal the vendor such as citations from manuals with stars (⋆⋆⋆⋆) and we hide the concrete procedures for hacking. If a vulnerability is found and the manufacturer cannot, or is not willing to cooperate we will contact the Dutch National Cyber Security Center (NCSC). The NCSC is a governmental organisation with as goal improving the security of the digital infrastructure[26]. Their website[27] provided us with a phone number: (070) 888 75 55 and an e-mail address: info@ncsc.nl which we could use to contact them.

### Assumptions

As described in the safety section we do not test on live systems. Therefore it is necessary to make the following assumptions. We assume that an explanted CRT device has the same functional characteristics as an implanted CRT device. We assume that the demonstration software from the pacemaker programmers has the same functional characteristics as the real program.

[26] `https://www.ncsc.nl/organisatie`
[27] `https://www.ncsc.nl/organisatie/contact.html`

### Safety concerns

To ensure that we only targeted the CRT device we are assessing and not accidentally the CRT device of a bystander, we reserved a big room with the possibility of locking the door. We ensured that the radio signal cannot reach the outside of the room and that non of the researchers had a CRT device himself. After the research we used the medical device destruction policy as provided by the hospital to dispose the device properly.

## 6.3. Device connectors and documentation

There are multiple ways to interact with the CRT device. For example the pacemaker programmer allows to reprogram the CRT device. The ⋆⋆⋆⋆⋆ however, only allows the transmission of patient data from the CRT to a central database. In this section we discuss the ⋆⋆⋆⋆⋆ and the ⋆⋆⋆⋆⋆. Notice that there are more devices that may connect to the CRT device we selected but that we do not discuss them all.

⋆⋆⋆⋆⋆

The⋆⋆⋆⋆⋆ is a device used to receive data sent by the CRT. The ⋆⋆⋆⋆⋆ is later used to transmits the data via a dial up connection to a healthcare professionals database. The documentation we


Figure 25: ⋆⋆⋆⋆⋆

analysed consists of two manuals: the healthcare professional manual ⋆⋆⋆⋆⋆ and the the patient manual ⋆⋆⋆⋆⋆. These manuals describe safety measures for operating the device and CRT device functionalities. We selected seven points from this manual that we think are interesting from an information security point of perspective.

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.

The first two statements we selected describe how the patient should use the device and what kind of behaviour should be avoided. During the security assessment the security assessor should verify that these statements are enforced by proper security mechanisms.

1. ⋆⋆⋆⋆.

This statement may indicate that there are security problems with interference on the radio frequency while transmitting.

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.

Statement four and five show that the device is connected via the public phone network. The security assessor should verify during the security assessment that appropriate security controls are in place to mitigate threat from external sources.

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.

The last two statements give insight in two procedures. The first procedure is the updating and inspection process which appears to be absent. The second procedure is about the disposal of the device. The security assessor should investigate what data could be extracted when the device is not disposed in a proper way.

| **Statistic** | **Value** |
|---|---|
| Expected Battery life time | ⋆⋆⋆⋆ |
| Decrement of battery life time on a non scheduled run | ⋆⋆⋆⋆ |
| Decrement of battery life time after shock | ⋆⋆⋆⋆ |
| Communication range | ⋆⋆⋆⋆ |
| Distribution of programmer device | ⋆⋆⋆⋆ |
| ⋆⋆⋆⋆⋆ Communication channel | ⋆⋆⋆⋆ |
| Update policy / options | The device does not require inspection or maintenance |
| Emergency protocol | No |
| Authentication | No |
| ⋆⋆⋆⋆⋆ Temperature limits Storage | ⋆⋆⋆⋆ |
| ⋆⋆⋆⋆⋆ Temperature limits operating | ⋆⋆⋆⋆ |
| ⋆⋆⋆⋆⋆ Input voltage | ⋆⋆⋆⋆ |

Table 10: Statistics collected during documentation analysis

### Pacemaker programmer

The pacemaker programmer is a device designed to be operated by healthcare professionals. The pacemaker programmer is able to read, write and modify therapy settings from the CRT device. From the interviews, as summarized in Table 11 we learned that the pacemaker programmer is


Figure 26: ⋆⋆⋆⋆⋆ [28]

a key device for accessing the CRT device and that it does not require authentication. We also learned that a pacemaker programmer is capable of programming lethal therapy. Although, the real problem is that the pacemaker programmer does not require authentication, we still found it interesting to discover where and how an attacker could obtain a pacemaker programmer. One of the healthcare professionals reported theft of a pacemaker programmer out of a healthcare professionals car. Theft could be a way to get hold of a pacemaker programmer. We also found that `www.ebay.com` offers 25 pacemaker programmers. We verified by a healthcare professional

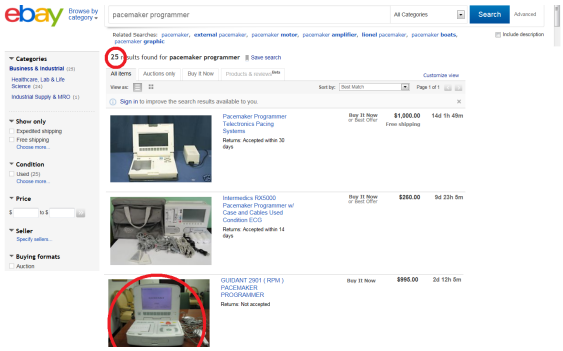



Figure 27: Pacemaker programmers

that at least two of the pacemaker programmers available on `www.ebay.com` are able to program pacemakers which are currently implanted in patients. Therefore we conclude that *Pacemaker Programmers, able to (re)program active implanted CRT devices are available to the public.* Most pacemaker programmer allow *USB access* and *Floppy access* and use them to backup CRT settings and patient data. We verified (procedure three and four) for the ⋆⋆⋆⋆⋆ that this data is not encrypted. Floppy and USB access is also used as a firmware / software update source by the manufacturer to update the pacemaker programmer. For analysing the documentation we used the healthcare professional manual [4]. In this manual we found four statements that indicate that the pacemaker programmer has the possibility for a direct WAN connection.

**Network statements:**

1. ⋆⋆⋆⋆.

[28] ⋆⋆⋆⋆⋆

2. ⋆⋆⋆⋆.
3. ⋆⋆⋆⋆.
4. ⋆⋆⋆⋆.

A direct WAN connection exposes the pacemaker programmer to external threats, for example hackers. ⋆⋆⋆⋆⋆⋆⋆⋆.

**Network Security statement:**

1. ⋆⋆⋆⋆.

The advice does not cover all aspects of information security to ensure a secure network infrastructure. A better advice would be to consult an information security professional. Two other statements indicate that the pacemaker programmer is capable of making a wireless connection.

**Wireless Network statements:**

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.

⋆⋆⋆⋆⋆⋆⋆⋆. As we learned from previous statements, the device supports a physical connection and is supposed to be on a secure part of the hospital network. Therefore one could question if adding WiFi support justifies the increment of possible WiFI attack vectors. The manual makes three statements we found interesting, about the pacemaker programmer its operating system and its default configuration.

**Operating System statements:**

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.
3. ⋆⋆⋆⋆.

In addition the manual makes three statements we found interesting about additional features the pacemaker programmer offers.

**Features and External Hardware statements:**

1. ⋆⋆⋆⋆.
2. ⋆⋆⋆⋆.
3. ⋆⋆⋆⋆.

Since USB sticks are easily lost, their use can increase the risk of leaking confidential data [35, 84, 51]. During the security assessment the security assessor should verify that there are appropriate counter measures [84] against data leakage via removable storable media. The second of these statements may indicate that ⋆⋆⋆⋆⋆⋆⋆⋆. The third statement indicates two things. The first is the availability of several diagnostic test. During the security assessment the security assessor should verify if the diagnostic tests are removed or that these procedures require authentication. The second part of the statement indicates that the device supports non-secure connections. This indicates a risk with respect to the confidentiality of the patient his or her data. Finally the manual describes how the device is activated.

**CRT Authentication statement:**

1. ⋆⋆⋆⋆⋆..

Since strong magnets are publicly available this should not replace proper authentication. By searching public sources we found four bug reports for the ⋆⋆⋆⋆⋆⋆⋆

## 6.4. Interviews

Even if an IMD is secure within an abstract model it may be insecure in a real environment. To get insight in some environments we tried to interview (12) different hospitals. Six healthcare professionals responded and some invited us to come and take a look at the programming process. To ensure that the healthcare professionals where able to answer our questions to their fullest knowledge we promised them that the results would only be used anonymously. In Table 11 we summarize the results of our interviews by statement.

| ID | Question | $RSP_1$ | $RSP_2$ | $RSP_3$ | $RSP_4$ | $RSP_5$ | $RSP_6$ |
|---|---|---|---|---|---|---|---|
| $Q_1$ | Programmer requires authentication. | no | no | no | no | no | no |
| $Q_2$ | Portability of the programmer device. | high | high | medium | high | high | high |
| $Q_3$ | Incidents from pacemaker theft from within the hospital. | 0 | 0 | 0 | $\frac{1}{2}$ | 0 | 0 |
| $Q_4$ | Employee is allowed to transfer pacemaker programmer by car. | yes | yes | ∅ | ∅ | ∅ | ∅ |
| $Q_5$ | Incidents pacemaker programmer theft from car. | 1 | 0 | ∅ | 0 | 0 | 0 |
| $Q_6$ | The pacemaker disallows certain input combinations. | yes | yes | yes | yes | yes | yes |
| $Q_7$ | Number of times that the responder got an unknown error message. | never | never | never | 1 per month | ∅ | ∅ |
| $Q_8$ | Pacemaker programmer stores a log that allows accountability checks. | no | no | no | no | no | no |
| $Q_9$ | Pacemaker allows logging. | no | no | ∅ | ∅ | no | no |
| $Q_{10}$ | The pacemaker logs therapy changes. | yes | ∅ | ∅ | yes | yes | no |
| $Q_{11}$ | The pacemaker programmer shuts down after some time. | no | no | no | no | no | yes |
| $Q_{12}$ | The pacemaker programmer terminates the connection with the pacemaker after some time. | no | no | no | no | yes | yes |
| $Q_{13}$ | Patients questioning the device its security. | many | some | yes | yes | some | some |
| $Q_{14}$ | Complains about electro magnetic interference. | very low | very low | no | very low | no | one |
| $Q_{15}$ | Is there a specific party or person responsible for updating and/or patching? | ∅ | yes | yes | ∅ | yes | yes |
| $Q_{16}$ | Responder verified our identity with a passport check. | no | no | no | no | yes | no |
| $Q_{17}$ | Pacemaker programmer maximum communication distance. | ∅ | ∅ | ∅ | 8 m | 10 m | ∅ |
| $Q_{18}$ | Pacemaker programmer allows USB use. | ∅ | ∅ | ∅ | yes | some | yes |
| $Q_{19}$ | Pacemaker programmer allows secure Internet connection. | ∅ | ∅ | ∅ | yes | ∅ | no |
| $Q_{20}$ | Pacemaker programmer allows CD use. | ∅ | ∅ | ∅ | yes | some | ∅ |
| $Q_{21}$ | Pacemaker programmer allows floppy use. | ∅ | ∅ | ∅ | ∅ | some | yes |
| $Q_{22}$ | Manufacturer needs password for software update. | ∅ | ∅ | ∅ | yes | ∅ | yes |
| $Q_{23}$ | The hospital verifies that the pacemaker programmer, programs it says it programs. | ∅ | ∅ | ∅ | no | no | no |
| $Q_{24}$ | Updates of the pacemaker programmer are documented. | no | ∅ | ∅ | ∅ | ∅ | no |
| $Q_{25}$ | Mentions risk of network virus to medical hardware. | yes | ∅ | ∅ | ∅ | ∅ | no |

Table 11: Quantified interview results, where ∅ means: not derivable from the interview.

## 6.5. Vulnerability assessment

### Capturing and channel observation

For our research we use the USRP as described in subsection 3.5. We extended the USRP with the WBX board rev 2 and WBX-FE-SIMPLE board rev 4. These daughterbords allow us to work with radio frequencies between 50 MHz and 2.2GHz. To avoid fast draining of the CRT battery, we decided to capture the communication traces between the CRT and the ⋆⋆⋆⋆⋆ and analyse the traces offline. Luckily, GNU Radio comes with a tool named: *usrp_rx_cfile.py* which takes samples from a frequency and writes these samples to a file. Because IMDs communicate between 402 MHz and 405 MHz we decided to start capturing everything around 403.5 MHz. After preparing the USRP we started a transmission between the CRT and the ⋆⋆⋆⋆⋆. We plotted the result with the WX GUI FFT Sink and shown the plot in Figure 28. This plot (Figure 28) has all

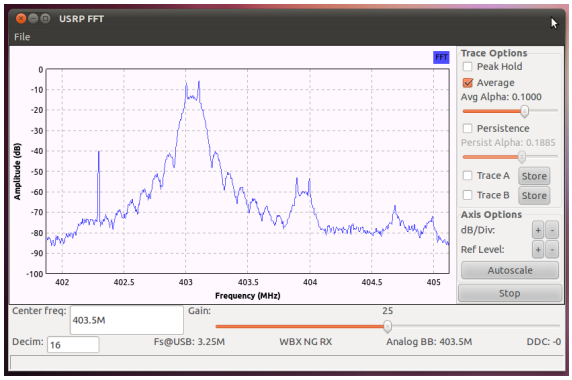



Figure 28: UHD FFT observeration

the characteristics of a communication channel as described by Fitzsimons [30]. We noticed that the CRT picks a frequency between 402 MHz and 405MHz at random. To capture a more clear trace we decided to display the frequency band between 402 MHz and 405 MHz. While displaying the band we started the communication between the CRT and ⋆⋆⋆⋆⋆. After observing the right channel, we started capturing the signal on the middle of the observed communication channel. By observing the communication channel via the WX GUI Scope Sink we found three different signals. In Figure 29, Figure 30 and Figure 31 we show these observations. From the beginning


Figure 29: UHD PSK observation

of the transmission up to approximately $\frac{4}{5}^{th}$ of the transmission we observed the wave pattern from Figure 29. Because Figure 29 seems to have a similar pattern as Figure 16 we assume that this part of the transmission uses a modulation scheme based on shifting the phase of the carrier wave. After approximately⋆⋆⋆⋆ of the transmission, we observed the pattern from Figure 30. This

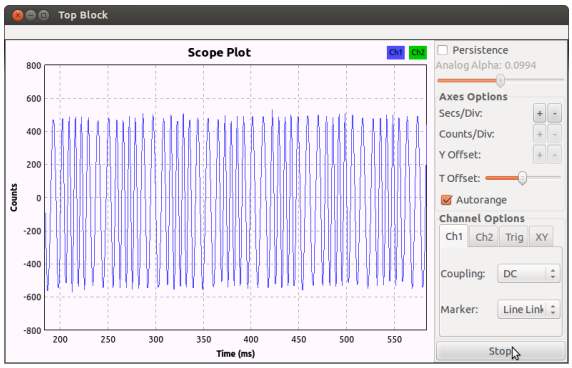



Figure 30: UHD FSK observeration

pattern looks very similar to the pattern from Figure 15. Therefore we think that this part of the transmission uses a modulation scheme based on changing the frequency of the carrier wave. The

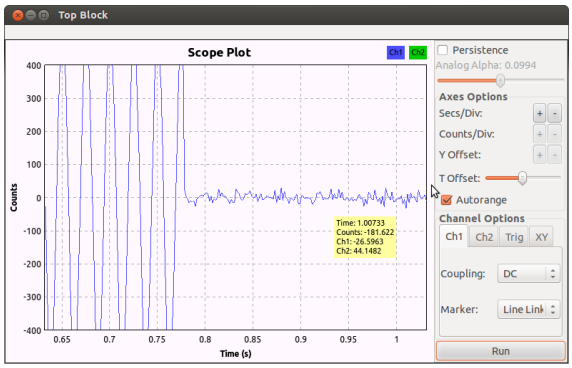



Figure 31: UHD carrier observation

transmission ends with a wave as shown in Figure 31. Which we assume to be the carrier wave mixed with noise. Since only medical devices are allowed on this channel and the channel is only observed during the transmission phase of the pacemaker we conclude that *the device existence privacy property of this IMD is not satisfied.*

The main part of the transmission uses a modulation scheme which changes the phase of the carrier wave. The simplest and most commonly used modulation scheme that uses the phase of the carrier wave is Phase Shift Keying. Therefore we decided to build a demodulation application as shown in Figure 32

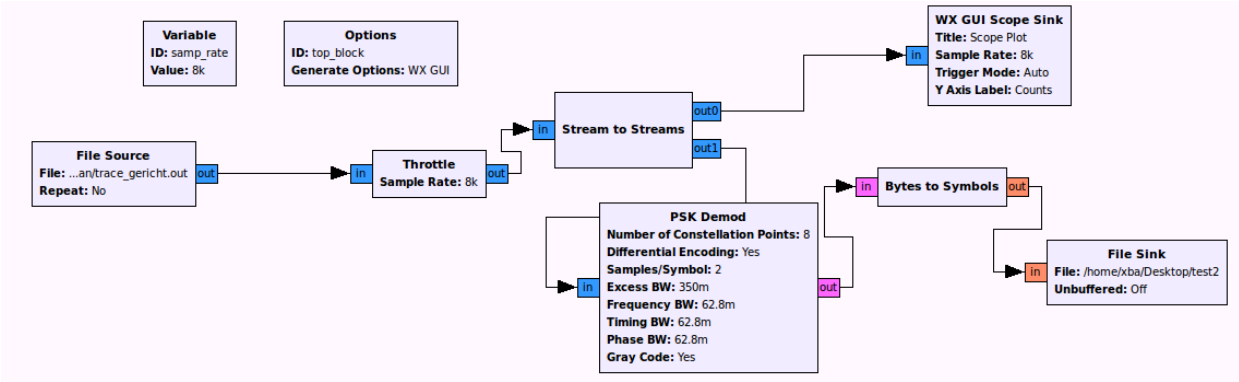



Figure 32: Demodulation attempt

The result of this demodulation scheme consisted of a repetitive hexadecimal pattern: 00 00 80 BF. This pattern is randomly interrupted by the hexadecimal value: 3F. Repeating this process with the DPSK demod sink results in exactly the same pattern.

### Access to the programmer device

One of the main methods for accessing the CRT device is via the pacemaker programmer. Each manufacturer has its own pacemaker programmer and each pacemaker programmer has its own characteristics. As shown in the Network Security statement above, the manufacturer relies on the security of the hospital for its security statement. As shown in Table 11 each hospital arranges the CRT programming process in its own way. Therefore the likelihood of successful exploitation of a vulnerability from a pacemaker programmer is depending on the security of the hospitals infrastructure. To estimate the risks, we first have to discuss some observations we found in the hospital environment and learned from the interviews. At the hospital where we did our experiment, the pacemaker programmers are stored in a room with the ability to lock. This room also accommodates a computer which is used to backup and view patients data. The backups are transferred via floppy or USB from the pacemaker programmer to the hospital patient healthcare system. This computer, which processes the pacemaker programmer backups is connected to the public internet. To assess the security of this computer we used the Personal Software Inspector (PSI). The PSI is a personal computer security solution that is able to identify vulnerabilities in third-party programs. These third-party programs and their vulnerabilities could leave your PC open to attacks. With the help of the administrator, we executed the (PSI) from Secunia[29]. We found that five software products where outdated, including Adobe acrobat and an Adobe Flash plugin. For both plug-ins Adobe acrobat (22)[30] and Adobe Flash (16)[31] we found that there are exploits available. After this finding the hospital took steps to ensure this cannot happen anymore. Based on this experience we wrote Procedure one. Procedure one describes a scenario to gain access to a computer which processes the pacemaker programmer backups.

**Procedure 1:**

1. Set up a website containing malicious code able to exploit vulnerabilities in the victims browser or browser plugins (called a drive by attack [20]).

[29] http://secunia.com/vulnerability_scanning/personal//

[30] http://www.exploit-db.com/search/?action=search&filter_page=1&filter_description=adobeacrobat&filter_author=&filter_platform=0&filter_type=0&filter_lang_id=0&filter_exploit_text=&filter_port=0&filter_osvdb=&filter_cve=

[31] http://www.exploit-db.com/search/?action=search&filter_page=1&filter_description=adobe+flash&filter_exploit_text=&filter_author=&filter_platform=0&filter_type=0&filter_lang_id=0&filter_port=&filter_osvdb=&filter_cve=

2. Lure the victim to the malicious website for example by sending him an e-mail.
3. Wait until the malicious code is activated and grants you a connection.
4. Place the pacemaker programmer malware or a backdoor for later usage.

Because of the missing security updates for Adobe acrobat and Adobe Flash (both available as browser plug in) *it is possible to compromise this system with a drive-by download.*
However, even when this system would be disconnected there is still a method to attack this system. For example, a method to gain access to the above system even when it is disconnected from the public internet could be a targeted social engineering attack. Procedure two describes a social engineering scenario to gain access to the computer processing the pacemaker programmer backups.

**Procedure 2:**

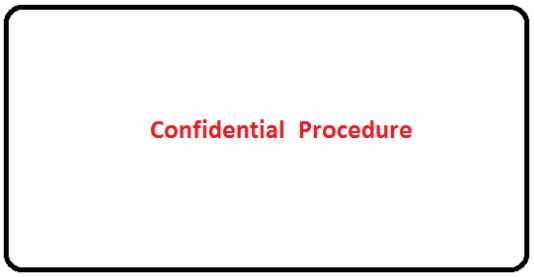
1. Preparing a USB stick with malware.
2. Hand the USB stick over to healthcare professionals just before they enter the pacemaker programming room.
3. Let this malware exploit local system vulnerabilities to gain more privileges and tamper with the system.
4. Place the pacemaker programmer malware or a backdoor for later usage.

If the malware is specifically designed for one target we speak of an Advanced Persistent Threat (APT) [21]. By evaluating the social engineering scenario *we deem it likely that an attacker could obtain access to this system.*

#### Attacking the demo application via the ⋆⋆⋆⋆⋆ programmer

The ⋆⋆⋆⋆⋆ programmer is able to program the ⋆⋆⋆⋆⋆ CRT device. We confirmed that the ⋆⋆⋆⋆⋆ programmer *does not require any authentication to be operated.* The programmer device contained a demonstration application with fake patient data. For safety reasons we only used the demonstration application. Procedure three describes the necessary steps to obtain a backup file.

**Procedure 3:** We used the USB stick and found a .⋆⋆⋆⋆⋆ file on it after running procedure



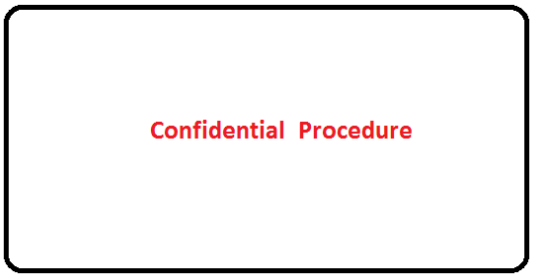

three. After the first backup, we changed one character in the patient name field via the ⋆⋆⋆⋆⋆ programmer and repeated procedure three. Both backup files where written to the USB stick and transported to our computer for further analysis. We opened the backup files in a hex editor[32] and observed that *the ⋆⋆⋆⋆⋆ programmer does not encrypt the patient data when transported to another device (USB or floppy)* As shown in Figure 33 we compared the two backup files[33] to

[32] `http://frhed.sourceforge.net/en/`
[33] ⋆⋆⋆⋆⋆


Figure 33: Memory difference of ICD programmer backup files

search for similarities. The first interesting observation is the fact that the ⋆⋆⋆⋆⋆ programmer makes a backup of the full memory instead of only the memory values. The memory consists of ⋆⋆⋆⋆⋆ segments ⋆⋆⋆⋆⋆. Segment ⋆⋆⋆⋆⋆ has ⋆⋆⋆⋆⋆ pages,⋆⋆⋆⋆⋆ has ⋆⋆⋆⋆⋆ pages and ⋆⋆⋆⋆⋆ has ⋆⋆⋆⋆⋆ pages. Each page has around ⋆⋆⋆⋆⋆ bytes of memory available which in total adds up to ⋆⋆⋆⋆⋆ bytes of memory. Because we only changed one character in the patients name and the backup comparing tool only displayed one deviation we found that the patients name is located at Segment ⋆⋆⋆⋆⋆. The hexadecimal value: "0x66" represents the letter "f' and the hexadecimal value "0x67" represents the letter "g". Procedure four describes the steps to modify and restore the patient name field via the backup file.

**Procedure 4:** After following procedure four, we found that instead of the letter "f" the let-




ter "g" was displayed on the screen. We conclude that: *the ⋆⋆⋆⋆⋆ programmer ⋆⋆⋆⋆⋆ demo application does not check the integrity of the backup.* Within the application the pacemaker programmer seems to allow only certain characters for the name input field. By replacing the "0x66" byte from procedure one, with a byte that represents a character that is not in that list, we where able to print the ç character on the screen. We conclude that *the ⋆⋆⋆⋆⋆ programmer⋆⋆⋆⋆⋆ Series demo application does not verify the name field from the backup for input restrictions* From the interviews we learned that the pacemaker only allows certain input for therapies. With this method it may be possible to program the pacemaker with a non-allowed or impossible therapy. This could be a threat to the patient safety. We observed that the length of the patients name was restricted. In memory there where ⋆⋆⋆⋆⋆ bytes available for the name parameter. When the name was shorter than the maximal available characters the memory space was filled with "0x00" bytes. We decided to test how the ⋆⋆⋆⋆⋆ programmer would react to a minimal overflow. Therefore we extended the ⋆⋆⋆⋆⋆ bytes long name memory space with an additional byte as shown in Figure 34. We found that this extension was pushed into the next segment. With other words the next parameter segment displayed the letter "g" as first character. *The ⋆⋆⋆⋆⋆ programmer ⋆⋆⋆⋆⋆ demo application does not check the back up input for memory soundness* or in other words the ⋆⋆⋆⋆⋆ programmer ⋆⋆⋆⋆⋆ demo application allows more memory as input than the pre defined structure allowed. Procedure five describes the necessary steps to cause a buffer overflow


Figure 34: Memory with additional byte of ICD programmer backup files

via the backup file

**Procedure 5:** Procedure five resulted in an error messages as shown in Figure 35.

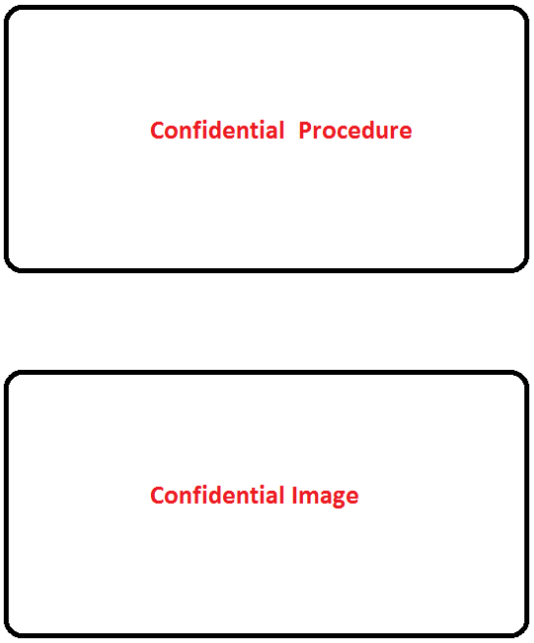



Figure 35: Programmer error message due to memory overload via CRT backup file

**Error message 1:**

1. ⋆⋆⋆⋆⋆

may also be trigger because ⋆⋆⋆⋆⋆ bug report[34]. To exclude this, we repeated procedure five with another arbitrary number of bytes. This resulted, as shown in Figure 36 a different error


Figure 36: Programmer error message 2 due to memory overload via CRT backup file

message.

**Error message 2:**

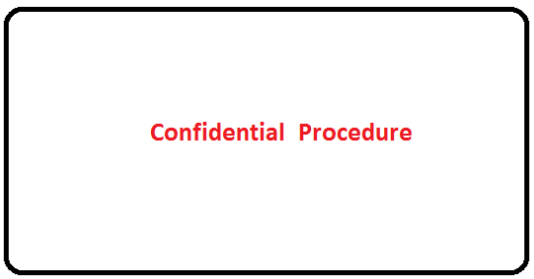
1. ⋆⋆⋆⋆⋆

Which is strange because the demonstration program should not be able to use ⋆⋆⋆⋆⋆. We also found that after following procedure five other input parameters in the program were removed. Therefore we conclude that *the ⋆⋆⋆⋆⋆ programmer⋆⋆⋆⋆⋆ demo application does not check the length of its backup file buffer* and may therefore be vulnerable for buffer overflows.

### Exploiting vulnerabilities of the ⋆⋆⋆⋆⋆ demo application

As argued in Section 6.5 an attacker may be able to obtain access to the computer that is between the pacemaker backups and the database. To exploit the vulnerabilities found in Section 6.5 an attacker could follow procedure six.

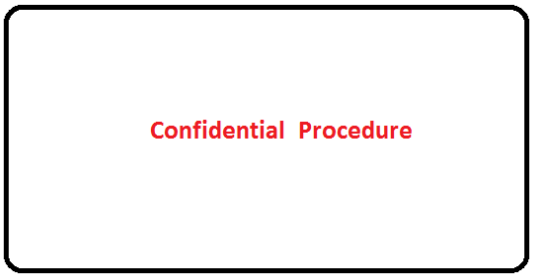
**Procedure 6:** This method could compromise the privacy of patients. To tamper with the




CRT device an attacker could follow procedure seven.

**Procedure 7:** A healthcare professional expects to click the program button for programming during an update. Therefore it may be possible to program the CRT device in a malicious way via the ⋆⋆⋆⋆⋆programmer. Notice that this way of exploitation focuses on the mass rather than a single target. The longer the malware is in place the more likely it becomes it makes a victim.

[34] ⋆⋆⋆⋆⋆




#### Comparing the demo application with a real application

Because we did not have a pacemaker programmer which would never be used afterwards any more, we did not test on a real application. However, to determine the probability that the attacks in the previous session would work on a real device we tried to compare the demonstration program with a real application. The real program works, just as the demonstration program with an USB stick and is able to write to the USB stick. We found that most real programs did separate read and write! There is one program to view previous pacemaker data from the USB stick and another program to modify therapy settings. The read program literately gave this error message on startup: ⋆⋆⋆⋆⋆ However, we found a program (as shown Figure 37) on the ⋆⋆⋆⋆⋆ programmer which allows both: reading, writing and programming. This program could


Figure 37: Program that does not separate read and write

be reached via: ⋆⋆⋆⋆⋆.

## 6.6. Threat analysis

Based on the manuals [1, 4, 2, 3] and the threats from the abstract IMD as listed in Table 7 we list the threats for the CRT in Table 12. Based on the interviews and the manuals [1, 4, 2, 3] we where able to model the environment from an IMD. This model is only applicable for one hospital and one home monitor box because other hospitals and patients could have a complete different setup. However, some parts are similar. The connection between the pacemaker programmer and the CRT does never differ and also the the connection between the CRT and the ⋆⋆⋆⋆⋆ does not differ. Based on this environment model and the threats from Table 12 we modelled the environment and the place of the threats in Figure 38

## 6.7. Impact assessment

In this section we report the findings from our vulnerability assessment according to the definitions in Section 7. Notice that all impact scores are worst case. For the attacks on application level we assumed that the ⋆⋆⋆⋆⋆, which does allow read and write, bears the same vulnerabilities as the demonstration program. Notice that finding three on itself has a low impact and that the

| ID | Threat | Description |
|---|---|---|
| $T_1$ | Unauthorized access to the CRT. | Allows an attacker to control the device. |
| $T_2$ | Data leaking from the CRT. | Personal information of the patients leaks from the device to the public. |
| $T_3$ | Malfunctioning / unexpected behaviour of the CRT. | The device its behaviour becomes unpredictable. |
| $T_4$ | Harmful behaviour of the CRT to the patients health. | The device its behaviour results in an event which is harmful to the patient. |
| $T_5$ | Denial of service of the CRT. | The device stops working. |

Table 12: Potential threats to the ⋆⋆⋆⋆⋆ CRT

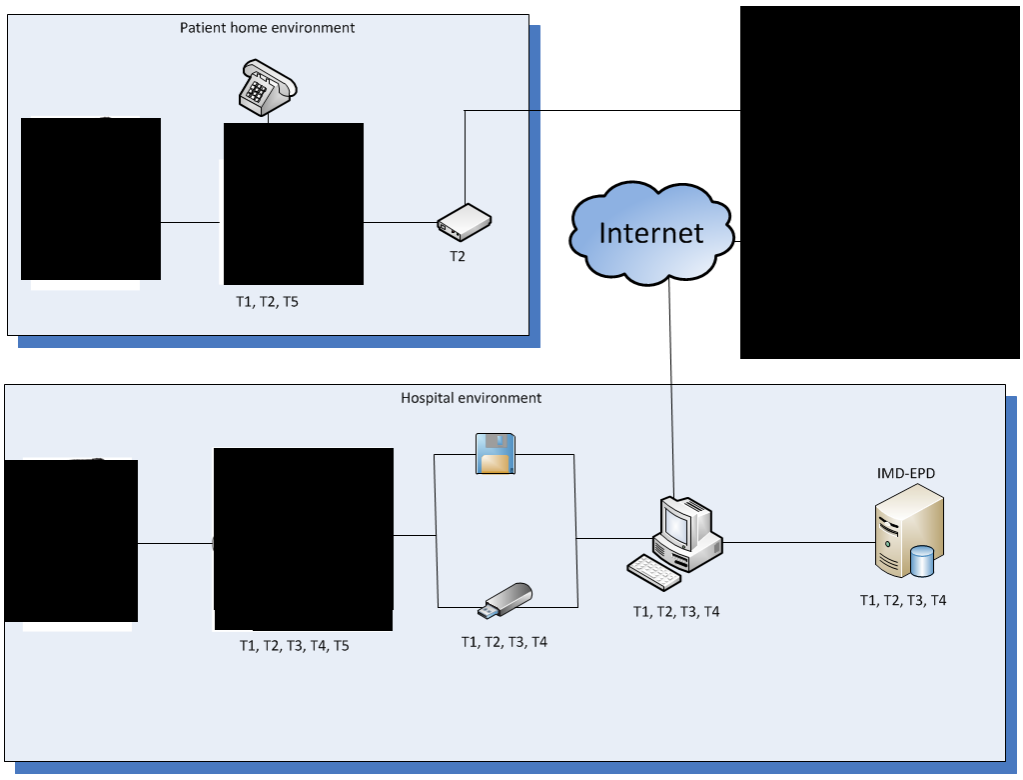



Figure 38: ICD threat location

real threat: lack of authentication is encapsulated by finding one. Another way to discuses the impact is via the security properties table from Section 4.4 In Table 6.7 we give an overview of the satisfied and the unsatisfied security properties desired for the ⋆⋆⋆⋆⋆ CRT. We deem the first four properties partially satisfied because under normal circumstances they will be satisfied. However, when we introduce a malicious actor they are not satisfied. From the 23 security properties we found that one is satisfied. We deemed five security properties partially satisfied. We deemed eight security properties not satisfied and for eight security properties we were not able to find if they are satisfied or not. When we count the partially satisfied security properties as half satisfied we conclude that less than a quarter of the desired security properties are satisfied.

| FID | Description | Impact | MASL |
|---|---|---|---|
| $f_1$ | The ⋆⋆⋆⋆⋆ Programmer does not require authentication nor for accessing it nor for accessing the IMD. Everybody with access to it may reprogram a patient. | Lethal | 3 |
| $f_2$ | IMD communicates over Medical Implant Communication Service frequency. It thereby reveals the device its existence. | Low | 2 |
| $f_3$ | Several pacemaker programmers are available on E-bay. | Low | 1 |
| $f_4$ | Computers processing IMD backup data have internet access and bear software vulnerabilities. | High | 2 |
| $f_5$ | IMD backup data is unencrypted. | High | 2 |
| $f_6$ | The ⋆⋆⋆⋆⋆ Programmer does not check backups for data integrity when consulted or re inserted. | Lethal | 3 |
| $f_7$ | The IMD does not have proper authentication. It is just a strong magnet in the programming head that actuates the sensor in the IMD. | Lethal | 3 |
| $f_8$ | The ⋆⋆⋆⋆⋆ Programmer does not check the length of the of an IMD backup. This could cause unexpected behaviour. | High | 3 |

Table 13: Findings and impact

## 6.8. Risk analysis

In the previous sections we described the threats and the vulnerabilities (findings) we found for the ⋆⋆⋆⋆⋆ CRT. According to the method as described in Section 7 we here calculate the risk corresponding to a threat. Notice that some vulnerabilities may reflect on multiple threats.

We found three vulnerabilities that reflect on the threat of unauthorized access to the IMD. The first vulnerability has a lethal impact and has a MASL of three. Therefore the *risk of unauthorized access is very high.* Since this is the highest possible classification and the highest should be the classification we are done for this threat.

We found six vulnerabilities that reflect on the threat of data leaking. Notice that each unauthorized access vulnerability also is a data leaking vulnerability since one could easy download all data when unauthorized access is obtained. The first vulnerability has a high impact and has a MASL of three. Therefore the *risk of data leakage is high.*

We found one vulnerabilities that reflect on the threat of unexpected behaviour. The first vulnerability has a high impact and has a MASL of three. Therefore the *risk of unexpected behaviour is high.*

We found three vulnerabilities that reflect on harmful behaviour. Because unauthorized access allows an attacker to use the NIST functionality[35] which is harmful, we counted three vulnerabilities which could result in harmful behaviour. The highest risk for unauthorized access is considered very high. Therefore the *risk of harmful behaviour is very high.*

We found no vulnerabilities that reflect on Denial of Service. Therefore the *risk of Denial of Service is low.*

## 6.9. Other pacemaker programmers

During our interviews we where able to view five different pacemaker programmers. We did not perform a full security assessment on them, as we did with the The ⋆⋆⋆⋆⋆ programmer. However, we found some interesting vulnerability indications which could be used as inspiration for further research.

[35] Section A

| PID | Property | is Satisfied |
|---|---|---|
| $p_1$ | Incident Treatment Delivery | Partially |
| $p_2$ | Necessary Treatment Delivery | Partially |
| $p_3$ | Right Treatment Delivery | Partially |
| $p_4$ | Incident Right Treatment Delivery | Partially |
| $p_5$ | Device existence privacy | No |
| $p_6$ | Device-type privacy | No |
| $p_7$ | Specific-device traceability | Partially |
| $p_8$ | Device data confidentiality | No |
| $p_9$ | Guaranteed emergency access | Yes |
| $p_{10}$ | Human Accountable Therapy Modification | No |
| $p_{11}$ | Device Accountable Therapy Modification | No |
| $p_{12}$ | Organisation Accountable Therapy Modification | No |
| $p_{13}$ | Authorized Healthcare Professional Device Traceability | No |
| $p_{14}$ | Authorized update source | Unknown |
| $p_{15}$ | Authorized IMD update | Unknown |
| $p_{16}$ | Authorized update | Unknown |
| $p_{17}$ | Time to update | Unknown |
| $p_{18}$ | Device data integrity | No |
| $p_{19}$ | Security breach notification | Unknown |
| $p_{20}$ | Attack recognition | No |
| $p_{21}$ | Standardized protocols and software | Unknown |
| $p_{22}$ | Self verification | Unknown |
| $p_{23}$ | User acceptance | Unknown |

Table 14: ⋆⋆⋆⋆ CRT security properties satisfaction

### The ⋆⋆⋆⋆⋆ programmer

After assessing the⋆⋆⋆⋆⋆ programmer we tried the same method on a ⋆⋆⋆⋆⋆r programmer. We confirmed that *the ⋆⋆⋆⋆⋆ does not require any authentication to be operated* The ⋆⋆⋆⋆⋆ programmer did not offer a demonstration program. Under supervision of the hospital we where granted the permission to make a backup from an existing explanted ⋆⋆⋆⋆⋆ pacemaker. The ⋆⋆⋆⋆⋆ pacemaker programmer offered the option to backup via an USB stick. It also offered many other backup options, we tried all the options. The ⋆⋆⋆⋆⋆ pacemaker programmer wrote four files (with extension:⋆⋆⋆⋆⋆) to the USB stick in a folder called: ⋆⋆⋆⋆⋆. The filenames where formatted according to the following pattern: ⋆⋆⋆⋆⋆. The ⋆⋆⋆⋆⋆ file contained *unencrypted data of a patient.* It also created a ⋆⋆⋆⋆⋆ file. When we made a name change in the ⋆⋆⋆⋆⋆ file it did not affect the ⋆⋆⋆⋆⋆ programmer. Even stronger, deleting the ⋆⋆⋆⋆⋆ file from the USB stick did not affect the possibility to import. We tried to ⋆⋆⋆⋆⋆ the backup ⋆⋆⋆⋆⋆ file with ⋆⋆⋆⋆⋆ [36], but it failed. We where able to view the file names inside the ⋆⋆⋆⋆⋆ file. The names had the same format as the patient file as described above. In addition, the ⋆⋆⋆⋆⋆ file contained ⋆⋆⋆⋆⋆ files with a different formatted name: ⋆⋆⋆⋆⋆. Because ⋆⋆⋆⋆⋆ did not work, we decided to view the ⋆⋆⋆⋆⋆ file in a hex editor. We observed that the backup file, as shown in Figure 39 is encrypted with . As argued in Section 6.5 an attacker may be able to obtain access to the computer which is between the pacemaker backups and the database. To exploit the vulnerabilities, as described above, an attacker could follow procedure eight.

[36] `$\star\star\star\star\star$`

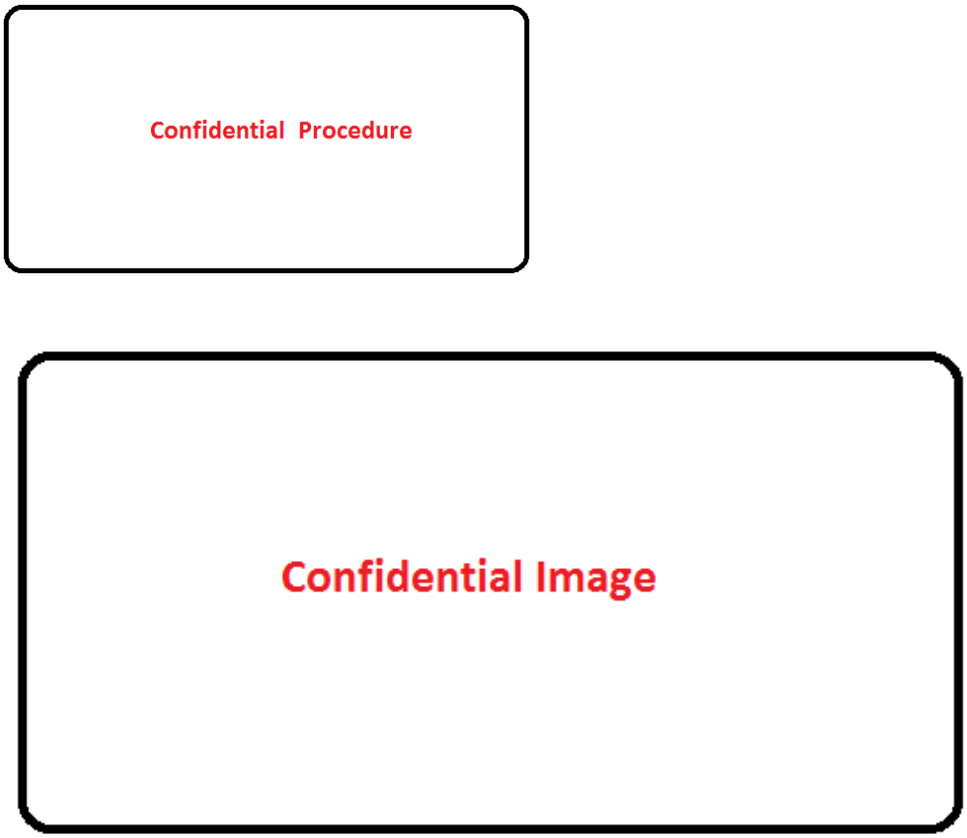
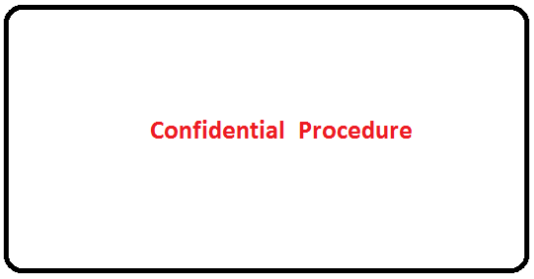



Figure 39: ⋆⋆⋆⋆⋆ backup file

**Procedure 8:** This method could compromise the privacy patients.




#### Test we did not do

Due to time constraints we where not able to do all the tests we had in mind. However, they may be interesting in further research. Therefore we list them below:

1. Tamper with the encrypted data of the ⋆⋆⋆⋆⋆ pacemaker programmer.
2. Try to obtain a firmware update file and try to tamper with it.
3. Try to eavesdrop on the dialup connection between the ⋆⋆⋆⋆⋆ and the central database.

4. Create a windows windows malicious filename extension via linux and upload that to the ⋆⋆⋆⋆⋆ programmer.
5. Re-authenticate until battery is empty.

# 7. IMD security assessment methodology

In this section, we propose a method for performing an independent security assessment on an IMD. This proposal is based on the applicable components from the security penetration testing standards in Section 5, the risk assessment from Section 4 and the practical considerations from Section 6. This method does not consider the safety of implanting the IMD nor does it consider the effect of the IMD material on the human body.

## 7.1. Planning and precautions

Before a security researcher starts a security assessment the security researcher should plan the security assessment and take some precautions. For the security assessor the planning should clarify the scope, device, and stakeholders. A project manager could extend the planning with project management controls but we did not include these in our methodology. The precautions should ensure that the security assessment is legal, ethical and safe. If the security assessment is not made on request of the vendor of the IMD, the security researcher should document how he is going to report and or disclose the findings. Finally the security researcher should document his assumptions because this substantiates the conclusion.

#### Scope determination

The first thing to do for the security assessor is determining the scope of the security assessment. The security assessor should for example, document if he is performing a blackbox or whitebox security assessment.

#### The device

To start a security assessment it is important for the assessor to know what is assessed. In this subsection the security researcher should describe the basic information about the device: name, manufacturing date, software version etc. The security researcher should also determine what devices or software is used to access the device.

#### Asset analysis

The first asset in the asset analysis of an IMD is the patients health / safety. Secondly, the device itself may be declared as an asset. The device may be an asset because it may be expensive or hard to replace. Something the device accommodates may also be an asset. For example, a drug infusion pump may contain very expensive medicines. An IMD might contain very personal information. Therefore also this data might be an asset. Every IMD may have its own unique assets. The enumeration of these assets make it more easy to determine the impact of possible vulnerabilities.

#### Stakeholders

In this subsection the security researcher should document all involved stakeholders. Stakeholders include: the patient, healthcare professional, hospital, pacemaker manufacturer, or insurance companies.

### 7.1.1. Ethical barriers

Within medical research ethical barriers are of great importance. One should first inform if the employer has an ethical committee. If so, one should adopt the applicable policies from this committee. Secondly, one should check if the customer has an ethical committee. If so, one should adopt the applicable policies from this committee. Finally the researcher could add his own ethical barriers to the research. Ethical barriers could include, for example:

1. Don't test on living patients.
2. Don't test on living animals.
3. Don't describe a lethal procedure.
4. Don't use patient data of a living patient.
5. Don't use patient data of a deceased patient.

### 7.1.2. Responsible disclosure

As described in subsubsection 3.6.6 patching and updating of the IMD could be a time consuming operation. Publishing (new) vulnerabilities in IMDs which are currently implanted and in use could result in major social unrest. Because of this possible major impact the IMD security analyst should document his responsible disclosure policy before he starts his research.

In the Netherlands the National Cyber Security Centrum published a manual about how to deal with responsible disclosure[37]. The first suggestion is to search for the responsible disclosure policy on the website of the manufacturer. It is always important to discuss how and when the vulnerability is communicated to the public. The NCSC prescribes that a hacker should act proportional and should not investigate further if he has enough evidence to prove his point. An example about proportionality is that it might be ok to view one database entry to prove a vulnerability, but it is unacceptable to download the full database. It is also not acceptable to use destructive or high damage impact tools.

### 7.1.3. Legal precautions

Depending on the scope of an IMD security assessment it might be necessary to document the legal conditions. Topics like: non-disclosure agreement and liability should be documented here. If any $3^{rd}$ parties are involved responsibility towards them should also be documented here. For example if an ISP is hosting a web environment it is necessary to inform them too in order to prevent a violation of the terms and conditions.

### 7.1.4. Set-up and Safety measures

It is unacceptable for an IMD to allow downtime on a working device. Nor is it acceptable to expose a patient, co-worker or other individual to a risk resulting from our security assessment. Therefore at least the following safety measures should be taken.

**Test on an ex-planted IMD** Make sure that the security assessment is performed on an ex-planted IMD. This is the most realistic environment considering possible modification from transplantation or modifications caused by activation.

**Test in a sealed environment** It is important that a security assessments on IMD is performed in a sealed environment. For safety reasons one should not assume that no others in the room have an IMD. To mitigate this challenge it is important to have a sealed room or environment to perform security assessments on IMDs.

**Safely end the assessment** It is important that the IMD is destroyed after the security assessment to ensure that no individual risks injury from a tested device.

### 7.1.5. Assumptions

Due to the safety measures we have to make certain assumptions for our security assessment. An important assumption is that an implanted IMD and an ex planted IMD behave the same. Also the use of fake patient data or demonstration software needs the behavioural assumption. For completeness we advise to document all assumptions.

[37] `https://www.ncsc.nl/actueel/nieuwsberichten/responsible-disclosure-uitgangspunt-voor-het-ncsc.html`

## 7.2. Execution

### 7.2.1. Gathering information

An important phase in a security assessment is the information gathering phase. During the information gathering phase a security researcher gathers as much information as possible/necessary. The security researcher could consult publicly available manuals, bug reports, public sources or interview relevant stakeholders.

**Documentation** An important part in determining the state of security of an IMD is reviewing the documentation. Documents about the architecture, publicly available specification (PAS), hardware and software could reveal useful information. For example statistics about: Battery life time, Decrement of battery life time on a non scheduled run, Communication range, Distribution of programmer device, Communication channel, Update policy / options, Emergency protocol, Authentication, Operating system, Max capacity (in the case of a pacemaker the max voltage of a shock and in case of the insulin pump: max amount of insulin per dose), Known errors, vulnerabilities and other bug reports In addition, threat models, bug reports and risk analysis about an IMD provide insight in the state of security of an IMD.

**Interviews** Not all information needed for the security assessment could be derived from the manual or the device itself. The device does not only need to be secure in its abstract environment but it must be secure in a real environment. Therefore it is important to interview all involved parties.

**Open sources research** A source of information could be open sources from the internet. The security researcher could search for internet fora which discuss the IMD. Youtube for movies about the IMD and Twitter for public tips, tricks and complains. Also the website of the manufacturer and advocacy organizations could provide valuable information and finally databases for scientific research may provide relevant material.

**Formal verification reports** In a proposal for amending the European Medical Devices: Directive 2001/83/EC[38] the European Commission mentions the obligation of (formal) software verification for the safety of software in ANNEX I. If these reports are publicly available the reports could be informative for a security assessment.

#### Vulnerability identification

Based on the scenarios from subsection 4.2 we created Table 15. The security professional should determine the applicability of each vulnerability for the IMD based on the documentation and interviews. During the interview and documentation phase it may be that some device specific vulnerabilities come up. The security assessor should add these vulnerabilities to Table 15. Now, the security assessor can assess all possible vulnerabilities documented to the IMD.

#### Vulnerability assessment

In the previous section we collected all possible vulnerabilities on the security assessment. In this section the security assessor should find out to what extend the vulnerabilities are applicable to the IMD. Based on the scope, blackbox or whitebox a security assessor is able to give more clearness.

**Authentication & access control and Emergency** For this method to be future proof it is important to verify authentication and access control. In the beginning the following security verifications will most likely result in a negative judgement. Because we found that IMDs currently do not have authentication. We propose that the security assessor should verify if:

[38] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0542:FIN:EN:PDF

| ID | Vulnerability Type | Results | compromises in CIA -triad |
|---|---|---|---|
| $V_1$ | Weak or non-existing authentication | Full device control | CIA |
| $V_2$ | Limited battery capacity | Battery exhaustion | Availability |
| $V_3$ | Wired communication | Denial of service | Availability |
| $V_4$ | Unencrypted communication | Eaves dropping<br>Communication manipulation | Confidentiality<br>CIA |
| $V_5$ | Weak encryption | Eaves dropping<br>Communication manipulation | Confidentiality<br>CIA |
| $V_6$ | Software / firmware vulnerabilities | Full device control | CIA |
| $V_7$ | Electromagnetic interference | Unexpected behaviour | CIA |
| $V_8$ | Radio Traffic analysis | Disclosure of confidential data<br>Full device control | Confidentiality<br>CIA |
| $V_9$ | Untrained / Unaware employees | Full device control | CIA |
| $V_{10}$ | Physical access | Disclosure of confidential data | Confidentiality |

Table 15: Vulnerability types for an IMDs

1. All resources and functions require authentication except those which are public.
2. Users can only access resources and functions for which they have the right level of authorization.
3. Public IMD functions cannot access non public functions or non public resources.
4. The set of emergency (public) functions is as small as possible.
5. Non emergency functions or resources depend on non public resources or non public function.
6. The set of patient (semi public) functions is as small as possible.
7. The authentication mechanism does enough to prevent a brute force attack but does not interfere with with the emergency possibilities.
8. The credentials of the authentication mechanism are sufficient to withstand a dictionary attack.
9. Users can safely change their credentials and do so from time to time.

**Battery** The security assessor should verify if:

1. The expected decrement of device life time, in case of an unexpected device connection or data transmission, is the same as documented.
2. The IMD notifies the patient when the battery is almost empty.

The security assessor should determine the possibility of a battery exhaustion attack. The security assessor could do this by discovering the most energy consuming function / part of the protocol. Than the security assessor should repeat this part / function until the IMD is empty and document the time. Based on the time and methods needed to repeat the function he could evaluate the risk of a battery exhaustion attack.

**Connection(s) and communication** For each connection an IMD could use, the security assessor should verify if:

1. The connection / signal does not reveal the IMDs existence.

2. The connection / signal does not reveal the type of the IMD.
3. The connection / signal does not reveal the location or the IMD owner.
4. The connection is encrypted.
5. The IMD only accepts one connection at a time.

Finally, the security assessor should document how the IMD connected with a device reacts to signal jamming.

**Software** The IMD itself is a dedicated device and therefore has only one application. For this application the security assessor should verify if:

1. A white list validation pattern is defined and enforced for all input.
2. Input rejection, rejects in a safe way.
3. A white list validation pattern is defined and enforced for all output.
4. Cryptographic modules, the cryptographic implementation and used algorithms are compliant with the latest NIST standard[39].
5. All confidential and critical data is tagged as such.
6. Only public data is disclosed to a public channel.

**Social engineering** The security assessor should verify that all relevant stakeholders are trained to be aware of social engineering attacks. Finally the security assessor should verify that the communication policies are as social engineering proof as possible.

**Physical security** If an attacker has physical access to an IMD it implies that the IMD is not inside the patient his or her body. Therefore the IMD is either ex-planted from a patient or on its way to implantation. The security assessor should verify for the ex-planted case that a proper destruction policy is in place, which is used by the healthcare professionals. In the implantation case the security assessor should verify that there exists a policy ensuring that there is not tampered with the IMD during the transportation phase.

**Logs and error handling** The security assessor should verify if:

1. Logs security related error messages thereby providing accountability.
2. Logs and error messages do not provide information to a user who is not authorized to get that information.

The security assessor should also document the maximal capacity of the log file.

## 7.3. Reporting and Evaluation

### 7.3.1. Threat analysis

In Section 4 we defined five threats for IMDs. After the information gathering phase the security assessor may have found more threats. In this section the security assessor should document them.

[39] `http://csrc.nist.gov/groups/STM/cmvp/standards.html`

### 7.3.2. Impact assessment

The impact of the successful exploitation of a type of vulnerability of an IMD may differ from IMD to IMD. We classify the impact: low, medium, high or lethal as described in Section 4. During the assessment the highest possible impact should be the classification. A finding is a vulnerability which is found during the security assessment. In Section 4, we defined five types of attackers: Non skilled attacker ($A_1$), Skilled attacker ($A_2$), Inside attacker ($A_3$), Funded organisation ($A_4$) and Nation states ($A_5$). We use these type to define the The Minimal Attacker Skill Level (MASL). The Minimal Attacker Skill level the first attacker in the list above which is able to exploit the finding. Since every step only increases the skill and resources it means that all attackers with the same or higher MASL are able to exploit the finding. The security assessor is now able to classify the impact for example according to the structure in subsubsection 7.3.2.

| FID | Description | Impact | MASL |
|---|---|---|---|
| $v_1$ | No authentication is requested. Everybody with a programmer device, available on E-bay could reprogram a patient. | Lethal | 4 |

Table 16: Findings and impact

### 7.3.3. Risk analysis

The final phase of our security assessment method is the calculation of risk. There are multiple definitions for risk. One method is to calculate risk as Impact x Probability. According to ISO/IEC 27001:2009, risk is calculated as Threat x Vulnerability x Impact. However, as we have stated before in Section 4, probability and likelihood is very dependable on the patient and his or her environment. Therefore the security assessor should calculate Risk for a threat t as: R(t) = riskLookup(Impact(vulnerability$_n$) x MASL(vulnerability$_n$)). Where the impact(vulnerability$_n$) is the impact of successful exploitation of vulnerability$_n$ and MASL(vulnerability$_n$) is the minimal attacker skill level needed to exploit vulnerability$_n$. With these parameters the security assessor could consult (meaning of riskLookup()) subsubsection 7.3.3 to find the corresponding risk. If there are multiple vulnerabilities which could activate a threat, the highest result will be the final risk score. The status Lethal is meaningful for the impact of a certain vulnerability. However, as risk this status is meaningless. To distinguishing between high and lethal in risk we call the successor of Lethal in the risk table: very high.

| | Lethal | High | Medium | Low |
|---|---|---|---|---|
| **MASL$_5$** | Medium | Medium | Low | Low |
| **MASL$_4$** | High | Medium | Medium | Low |
| **MASL$_3$** | Very high | High | Low | Low |
| **MASL$_2$** | Very high | High | Medium | Low |
| **MASL$_1$** | Very high | High | Medium | Low |

Table 17: Risk lookup table

# 8. Security recommendations

In this section we give eight recommendations for the improvement of IMD security. These recommendations are based on our experiences from the interviews, security assessment and literature study. We did not experiment with these recommendations in practise and can therefore make no statement about the clinical effectiveness of these recommendations. Also notice that these recommendations are examples of recommendations. They might be more or less effective depending on the type of IMD you are improving and every IMD may require deeper-measures as well.

## 8.1. Security recommendations for the IMD

In this section we discuss three security recommendations for the IMD. The IMD product lifecycle can extend over a 20 year time span [13]. Because these recommendations focus on this lifecycle, implementation and deployment of these recommendations could take up to 20 years.

### Second channel Notification

The implementation of this recommendation could mitigate the risk of unauthorized action and focuses on the desired security properties: "Security breach notification" and "Attack recognition" as defined in Section 4. Some IMDs are able to provide a sound, for example when the battery is almost empty [39]. We could use this functionality for a security improvement. For example, when a connection is established the IMD gives a beep. If the connection is made at an unexpected moment the patient could walk away and call a healthcare professional to check the settings from the device. The downside of this security improvement may be that playing a sound is energy consuming, making an attack on the limited battery more likely to succeed. Further the beep should not have false positives, since the stress caused by the fact that there could be something wrong with the IMD is not good for the patient his or hear health. Finally there is a risk that an attacker records and replays the sound or that an environmental sound may sound similar. This false positive risk could be mitigated if the patient is able to verify if the settings of his IMD have changed since the last reprogramming of his IMD.

### Segregated batteries

The implementation of this recommendation could mitigate the risk of Denial of Service and focuses on the desired security properties: "Incident Treatment Delivery" and "Guaranteed emergency access" as defined in Section 4. As seen in subsection 3.1 most IMDs use one battery for both communication and the critical process. The risk of an successful attack on the availability [80] could be mitigated by creating two circuits. One circuit is for the telemetry and could be compromised by an attack. The other circuit should control the essential process: data acquisition and therapy delivery. In addition this circuit could run a new process that checks the working of the telemetry and battery at some interval. Than, if the the telemetry battery does not respond, the battery for the essential process should invoke a second channel notification.

### Self powered IMDs

The implementation of this recommendation could mitigate the risk of Denial of Service and focuses on the desired security properties: "Incident Treatment Delivery" and "Guaranteed emergency access" as defined in Section 4. If the concept of a self powered IMD battery [40] comes to market it could improve the security with respect to all desired security properties. The battery vulnerabilities will be far less effective and the dive could use more powerful cryptography. This makes it possible to use stronger communication protocols and encrypted confidential IMD data.

[40] http://article.wn.com/view/2012/09/04/A_selfpowered_pacemaker_with_no_battery_coming_soon/

## 8.2. Security recommendations for the IMD healthcare professional equipment

### Authentication for non emergency privileges

The implementation of this recommendation could mitigate the risk of unauthorized action and focuses on the desired security properties: "Guaranteed emergency access" and "Human Accountable Therapy Modification" as defined in Section 4. From the interviews we learned that the availability of the device is of a high priority. It is not acceptable that a patient dies because a healthcare professional is not able to access the IMD. Nor is it acceptable that a healthcare professional loses time by authenticating himself to a programmer device in case of emergency. However, not all functionalities are necessary in case of emergency. For example, restoring a backup or running a test procedure is non critical emergency functionality. A major improvement to IMD security can be accomplished by the use of user accounts. One emergency (default) account, which only allow functionality that is needed for saving a patients life in case of emergency. The second account requires authentication from a healthcare professional but has access all functionality.

### Risk zone non reprogrammable

The implementation of this recommendation could mitigate the risk of unauthorized action and focuses on the desired security properties: 'Organisation Accountable Therapy Modification" and "Human Accountable Therapy Modification" as defined in Section 4. For some devices it may be feasible to divide functionality in multiple groups. One group for emergency functionality, one group for normal usage and one group for reprogramming the device. The normal usage and reprogramming group should only allow requests when send from within a specific location. Based on the GPRS locations, an IMD programmer is only allowed to operate in a certain environment [95]. This solution looks promising but, as we now from the interviews, there are healthcare professionals who transport the IMD programmers to the patient. Therefore limiting the possibility to program within a certain area may conflict the "Guaranteed emergency access" property.

### Biometric authentication

The implementation of this recommendation could mitigate the risk of unauthorized action and focuses on the desired security property: 'Guaranteed emergency access" as defined in Section 4. Many papers state the problem with passwords as an emergency access enforces. One of the problems is (emergency) access to the IMD in the case that the patient is unconscious. A solution proposed for this problem was a tattoo. However, many patients did not like the idea of having a tattoo. The use of Biometrics may be a solution for this problem. Even if the patient is unconscious, the patient biometrics are available and could be used by the healthcare professional to run the authentication process.

## 8.3. Security recommendations for the IMD infrastructure

### Accountability

The implementation of this recommendation does not mitigate a specific risk. The IMDs we have seen did not have any option for accountability. Even without the concept of user authentication one could start with the implementation of accountability. Due to legislation each IMD should have an Unique Device Identification[41]. This identifier could be used to log changes made by the connected device. The log file for therapy changes should be a cyclic log file. Each time the IMD patient visits the healthcare professional for a follow up session the log file should be downloaded, checked and cleared. The IMD is able to store parameters. Instead of the time parameter, which is not available on an IMD, an event parameter could be used to have some chronological order in the log file.

[41] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0542:FIN:EN:PDF

#### Use an IMD security assessment methodology

The implementation of this recommendation does not mitigate a specific risk. In the long term it may contribute to mitigate all risks. For software security evolution it is important to receive continuous feedback [8]. Important in information security is that information security is not a product but a process [17]. By using our proposed security assessment methodology an IMD manufacturer could continuously evaluate his product.

# 9. Conclusion and Discussion

To provide an answer to the problem statement, we start by answering the research questions we stated in Section 1. Then, we answer the main question and finally suggest topics for further research.

## What security properties are desirable for an IMD?

In Section 3.1 we defined an abstract IMD. This abstract IMD consists of four components: Data acquisition component, Telemetry Unit, Configuration Component and the Action component. We argued why this abstract architecture is correct and compared it with the architectures of a Deep Brain Stimulator, a Generic Insulin Infusion Pump and a pacemaker. Our abstract architecture encapsulates all three architectures we described. Based on the abstract architecture and characteristics from Section 3.1 we enumerated the desired security properties for an IMD. We defined 23 desired security properties for an IMD and categorized them in seven classes: Therapy safety, Privacy, Emergency access, Accountability, Detection and verification, Patching updating and incident response and Other. We found these properties useful for determining the current status of IMD security, rating the impact of the possible exploitation of vulnerabilities, reasoning about several attacks and vulnerability classifications. Although, satisfying these security properties should improve safety, it might affect the clinical effectiveness of the device [71]. We did not experiment with our security properties in such a way that we could make a statement about the clinical effectiveness of the device.

## How can we attack an IMD?

In Section 4 we classified five types of attackers: non-skilled attacker, skilled attacker, inside attacker, well funded organisation and nation states. We also classified ten types of vulnerabilities: Weak or non-existing authentication, Limited battery capacity, Wired communication, Unencrypted communication, Weak encryption, Software / firmware vulnerabilities, Electromagnetic interference, Traffic analysis, Social engineering and unsecured physical access. For each vulnerability type we described the corresponding attack method and rated the necessary skill level that an attacker needs to have to execute the described attack.

## How applicable are the current methodologies for IMD security testing?

In Section 5 we rated two popular security standard for their applicability for IMDs. We deemed the OWASP security verification standards more than half applicable to an IMD security assessment and we deemed the SANS security controls a bit less than half applicable to an IMD security assessment. We also used the risk assessment methodology from ISO/IEC 27001:2005. In Section 6 we found this method very applicable for our risk assessment methodology. In the end we conclude that current standards are partly applicable to the IMD setting to carry our a security assessment.

## What is the current status of IMD security?

As we have seen in Section 6 only 15.2% of our desired security properties where satisfied for a randomly chosen IMD on which we performed a security assessment. We found several vulnerabilities for this IMD and concluded that risks of unauthorized access and the risk of harmful behaviour for this IMD and according to our definitions from Section 7 is very high. We also found that the risk of data leakage and unexpected behaviour for this specific IMD is high. In total we found four risk with a classification of “high” or “very high” for this randomly chosen IMD. Although, we contacted a limited amount of vendors and only performed a security assessment on one IMD, we think that it is not unlikely that there are more IMDs with the same risks as we have described above.

### Can we mitigate IMD security risks?

In Section 6 we demonstrated that our security assessment methodology, as described in Section 7 was able to identify some weaknesses in an IMD. As with other systems, penetration testing could be part of a product development life cycle. In Section 8 we gave eight security recommendations to mitigate security risks. Not all recommendations are applicable to all IMDs and some recommendations require global change of the healthcare infrastructure. For example, if we want to be compliant with emergency access requirements [23] and we want to use an authentication mechanism, we have to deal with key management on a global level, which is challenging [71]!

### How can we improve IMD security?

There are techniques, used in security engineering that could be used to analyse and assess the security of IMDs. As we have seen, IMDs differ in security requirements and in characteristics from other systems. We found that only parts of currently available security standards are applicable to IMDs. We demonstrated that, by applying parts of these standards, we where able to identify vulnerabilities for a randomly chosen IMD. However, that an IMD security assessment methodology is able to find IMD vulnerabilities does not say that it is able to find all IMD security vulnerabilities. In other words, we cannot guarantee that the method is complete. This also holds for the security recommendations. We think they may contribute to a considerable part of the solution but do not provide a solution to the whole problem. Subject to these conditions, we think that currently existing security methodologies are able to give a considerable contribution to the security of IMDs .

## 9.1. Further research

Due to time constraints, we did not do every possible experiment we had in mind. For example we did not try to exhaust the battery or jam the communication between the CRT and the ⋆⋆⋆⋆. Security researchers could try to develop a method to do this in a standardized way. For the wireless communication, the IMD security assessment methodology could be further refined. As shown in Figure 29 and Figure 30 and, as discussed in Section 6, the IMD communication pattern changes at some point during the communication. Researchers could aggregate trace information of several IMD types and see if an unique pattern could be derived (for example by statistical analysis) for each device type. Finally we showed that, if we want to implement authentication for IMDs, it is necessary to investigate how the global key management [71] for IMDs could be arranged.

Another area in which IMD research could proceed is the cross disciplinary area. For example in the policy and legal direction. Because it takes much more time to update all IMDs than to update a single standardized information system, normal responsible disclosure guidelines may not be sufficient for the IMD setting. We realize that we only made a small start with our responsible disclosure policy and think it might be a nice topic for further cross disciplinary research. Another cross disciplinary research topic could be to find out what the social acceptability of our security recommendations are.

# A. Healthcare professional survey

To get insight in some environments we tried to interview (12) different hospitals. Six healthcare professionals responded and some invited us to come and take a look at the programming process. To ensure that the healthcare professionals where able to answer our questions to their fullest knowledge we promised them that the results would only be used anonymously.

## Questions

1. Does the programmer device requires any authentication (login with username and password) to operate?
2. How would you qualify the level of transportability of the IMD programmer (IMDP)?
3. How do you deal with old IMDP's?
4. When not operated, where are the IMDP's stored?
5. To what extend does the software of the IMDP tries to prevent wrong input, negative numbers, extreme high values etc?
6. How often do you experience undefined errors from the IMDP?
7. Does the IMDP provide audit trails for accountability?
8. Is there a maximum operating time after which the IMDP shuts itself down?
9. is there a maximum connection time after which the IMDP suspends the connection with the IMD?
10. Do patients question the security of an IMD?
11. How often do you receive complaints about electromagnetic interference?

## Answers

### RSP1

The $1^{st}$ responder told us that the pacemaker programmer does not require any sort of login. It starts up automatically and does not ask for any information. The responder mentioned that they must be able to act fast in case of emergency but he mentioned that an emergency case probably needs less available functionality. The responder also mentioned that this could be a risk within the social circle, a pacemaker programmer expert could for example exit his marriage in this way. The pacemaker programmer felt like it was designed to be portable. But when the pacemaker programmer was not in use, it is stored in a so called: 'pacemaker control room'. The responder did not remember any theft incident from that room. However the responder did mention an incident in which a pacemaker programmer was stolen from within a car. But ensures us that the pacemaker programmer is useless in the hands of the average person. The pacemaker programmer is classified as "average risk hardware" by the hospital and destroyed as described in the hospital procedure. The pacemaker programmer remains property of the manufacturer which also is responsible for updating and patching. The responder mentions that the patching process does not involve any specification from the manufacturer. The pacemaker programmer is just patched without any information what part of the programmer is patched, what functionality changes etc. With this in mind, the responder makes a remark about a study by Hauser [42] about the possible death caused by programming faults and also mentions that there are stories about deceased patients by which, after inspection the pacemaker was disabled. The responder mentioned that the pacemaker programmer had limited configuration options and disallows certain combinations of them. It is however possible to program the pacemaker in the wrong way. For

example there is a procedure that forces a pacemaker to bring the heart in a certain state. This procedure is used to test if the sensors are reacting / working on the right signals. Therefore only specialist healthcare professionals are allowed to operate the device. There is however no logging available on who made the modification. The pacemaker does log some information about previous therapies. Some pacemaker programmers could backup data from the pacemaker to a floppy disk. This floppy disk could then be used to transfer the data to an electronic patient system. The responder mentioned a story about a USB virus infection of other medical hardware at some hospital a while ago. The responder almost never got unexplainable error messages on the programmer device and mentioned that this even holds stronger for newer programmers. There are no error messages for a maximal connection time after which the connection with the pacemaker is broker nor does the pacemaker programmer shut down after a (fixed) period of time. The responder told us that a there are many patients who ask questions about the security of the pacemakers. But that he never got any complains about electro magnetic interference.

### RSP2

The $2^{nd}$ responder replied that the pacemaker programmer does not require authentication. It was deemed portable by a transport car or hand since its weight is around 5kg. This hospital does not own the pacemaker programmer but hires it from the manufacturer. This manufacturer deals with maintenance and updates. When the pacemaker programmer is not in use, it is stored in a locked room. The risk of theft is comparable to all other locked rooms in the hospital. The employees from the manufacturer are allowed to take the pacemaker programmer in their cars for transport. The pacemaker programmer disallows certain combinations. For other combinations it just gives a warning, just as it gives a warning for negative or real high numbers. The responder never had an unexplainable warning. The responder told that the pacemaker programmer does not shut down automatically, so if there is a connection with the pacemaker it remains until the healthcare professional ends it. The responder replied that the pacemaker programmer does not provide any accountability. His patients ask about the security of the pacemaker on occasion and did not experience bad event with electromagnetic interference.

### RSP3

The $3^{rd}$ responder, responded that the pacemaker programmer does not require authentication but that he would be interested in an experiment to create a login on the programmer and then count the number of patients that die during an acute incident due to the fact that a random person cannot use the pacemaker programmer (we assume the responder was sarcastic on this one). The pacemaker programmer is highly portable but attached to a cart. When the pacemaker programmers are not in use one is stored on the catheterisation room and the other on the polyclinic. The manufacturer is responsible for the software updates of the pacemaker programmer. The manufacturer is obligated to ensure that every pacemaker ever created by the manufacturer is supported by any new pacemaker programmer. The pacemaker programmer gives a warning when a wrong setting is inserted. This setting then cannot be programmed to the pacemaker. The pacemaker programmer never gives an unexplainable error and does not register the name of the operator nor does it registers the actions which where performed. As far as the responder knows the device does not shut down after a specific amount of time. He ensures us however that he always shuts it down after operating it. He does not know if there is a maximum connection time from the pacemaker to the pacemaker programmer. The responder states that patients often question the security of the pacemaker. They also have questions about electromagnetic interference, but in practise they do not have complaints about it.

### RSP4

The $4^{th}$ responder replied that the pacemaker does not require authentication but that they are vendor depended. He noticed that a password would be very inconvenient during an emergency

situation. A pacemaker from ⋆⋆⋆⋆ cannot be reprogrammed by a pacemaker programmer from ⋆⋆⋆⋆. Some time ago an attempt was made to build a universal pacemaker programmer but the experiment failed. The responder told that not every pacemaker programmer could program each previous pacemaker of that vendor. Therefore they still have some old pacemaker programmers. The responder stresses that due to the new communication options the care of the patient definitely improved. The pacemaker programmer is limited to 8 meters and needs a protocol activation from a very near distance. He states that in theory there may be a probability that you could reprogram the ⋆⋆⋆⋆ set to follow a different protocol. But the probability for this is very low. The responder states that the device is made to be portable. It is like a laptop with a handle. This is necessary for a hospital to be as fast as possible on a place of emergency. The manufacturer is responsible for the updates and destruction of the devices. The hospital is responsible for the extrema, for example if a pacemaker programmer drops hard on the ground. Once in a while the manufacturer drops by at the hospital and updates the software of the pacemaker programmer via USB or a CD. The manufacturer has to insert a password before the software update may be performed! In some special cases the pacemaker programmer could make a secure internet connection to acquire a software update. But it is normally performed by an employee of the manufacturer. The responder deems the probability of theft of a pacemaker programmer low. The pacemaker programmer is in the pacemaker room most of the time. This room is either locked or crowded with healthcare professionals. However the responder mentions that he knows a story from another hospital in which a repair man told the healthcare professionals that he was there to repair the TV. This repair man took the TV and never brought it back. Although the manufacturer is responsible for the destruction of old devices the responder mentioned that during the transmigration of the hospital some very old pacemaker programmers where thrown away together with the normal garbage. The pacemaker programmer gives a warning when two contradicting parameters are selected and parameter or therapy selection is done via drop down menus. The only field on which text could be inserted is the name field. This name is limited to some amount of characters depending on the device type. The responder mentions that the pacemaker programmer sometimes freezes. This happens by approximative once in a month. Another warning that seems to pop up once in a while is the 'wrong handshake' warning. The pacemaker programmer does register the modification in therapy but does not provide any accountability. The responder then mentions that he always assumes that the pacemaker programmer does whatever it says it does but that they did not verify a pacemaker programmer actually does what it says it does. The connection or pacemaker programmer shut down depends on the healthcare professional. It does not automatically switch off, which is good since the time of programming a pacemaker varies by therapy and healthcare professional and it should not disconnect or shut down when the pacemaker is still being programmed. The responder mentioned that the patients always question the safety of the pacemaker but he never had a patient questioning the hackability of the device. He then mentions a story about a patient which did not tell his dentist he had a pacemaker. When the dentist started the nerve therapy the pacemaker concluded that there was something wrong with the heart and delivered a shock to it. The manufacturer does not give a guarantee against interference.

### RSP5

The $5^{th}$ responder invited us to the pacemaker programmer room to find answers by observation. After identifying ourselves by passport and college card the responder showed us the pacemaker programmer room. The pacemaker room contained many device types and was locked. The devices however are portable but except for one type of programmer, it would be too obvious to just walk away with it.

We observed that the pacemaker programmer was accessible by a Floppy disk. We observed that the pacemaker programmer was accessible by a USB stick. Via the USB stick it is possible to backup patient data. The data, nor the USB stick where encrypted. Data could be modified in a normal computer and in normal circumstances placed back on the computer. But we did not verify if the data was afterwards accepted as legitimate backup.

The responder mentioned that other vendors do have the option to back up the settings of the ICD with the possibility to place the old backup settings back to the pacemaker PROGRAMMER. Never back to a pacemaker!

We observed that the pacemaker programmers questions, if the connection is still necessary / if the patient if still in the room, after several minutes. The pacemaker connection could only be initialized when the pacemaker programmer head button was placed on the ICD. After the head button did the handshake, the head button was not necessary anymore. The responder told that he some times experiences a connection locks when in the other room a patient's ICD was reprogrammed.

Therefore it is not possible to use a wireless transmission within 10 meters with several ICD's (security measurement).

We also tested the range of the connection. We established a connection with the ICD and then walked backwards until the pacemaker programmer displayed the connection lost message. After +/- 10 meters we got this message. The responder mentioned one case in which a patient questioned the ICD security, e.g. responded to a news article that claimed ICD where hacked (tweakers.net)[42] The pacemaker programmer could be capable of producing a possible lethal scheme. However this scheme could only be executed via the pacemaker programmer and cannot be programmed on the ICD. These schemes exist for testing purposes, for example to see if the trigger of the ICD works correctly.

It is interesting to mention that except the patient name and address, the therapy programming is done by preprogrammed values. Within the GUI only a pre specified set of therapies is possible to program. For the other variables we observed that it was not possible to insert bigger values then reserved for the text variables via the GUI. Another interesting option the responder mentioned was the fact that it is possible for healthcare professionals to view the status and some minimum patient information (Patient + patient number + ICD type) via the internet and via their mobile phone. This does require authentication from the healthcare professional to the website (and a secure connection; https).

### RSP6

The $6^{th}$ responder invited us for an interview in the hospital. In this hospital only the pacemaker programmer professionals are allowed to operate the pacemaker programmers. Meaning that most healthcare professionals are not allowed to operate the device. They start each session clean, meaning that they read the data from the pacemaker in the system. They do not revert backups from the EPD to the pacemaker programmer. However it may be that a pacemaker programmer professional wants to revert to an old therapy. Then the pacemaker programmer professionals opens the EPD and physically inserts the old settings back to the pacemaker programmer. There is however no other connection than the human connection between these systems.

They showed us different pacemaker programmers all without authentication. They also mentioned that for the web application which processes parts of the ICD data there is no password change policy by the two biggest vendors: ⋆⋆⋆⋆ and ⋆⋆⋆⋆. Boston scientific asks for another password to access the web application every 6 months. Via the contract, the responder is assured that the data does not leave the European Union. The database of the web application is located in England.

The responder tells that all therapies are selected from a drop down menu. However, some functionality called NIPS bears more risk than other functionality. The NIPS functionality is able to end an arrhythmia. However, with this functionality, a patient without a Cardiac dysrhythmia could be put in a status in which he gets one. Another interesting option is the MRI modus. This modus is only available for the ⋆⋆⋆⋆ programmer. When the MRI modus is programmed in the pacemaker, each healthcare professional with an ⋆⋆⋆⋆ activator[43] is able to change the patients

[42] http://tweakers.net/nieuws/85021/onderzoeker-legt-ernstig-beveiligingsgat-in-pacemakers-bloot.html, http://tweakers.net/nieuws/52395/hackers-kunnen-inwendige-defibrillators-op-afstand-bedienen.html

[43] ⋆⋆⋆⋆

his therapy for 90 minutes.

The pacemaker does not use the sensor during this modus. This could be a risk when the patient get nervous. His heartbeat will increase but the pacemaker will not notice this and still pace in the programmed way. Which could result in a very unhealthy high hearth rhythm. Another risk is that this MR status device is very small (as shown in Figure 40). This increases the risk of theft


Figure 40: ⋆⋆⋆⋆ activator

and the risk of successfully attacking the patient. For this attack to work, the attacker should be able to be in a 10 cm range from the victim. A final interesting setting is the second channel notification. In newer versions it is possible to disable this setting. The second channel notification is an audio signal which occurs for example if the battery is almost empty.

The pacemaker programmer is updated via a representative of the vendor. Before the vendor is able to update the firmware he must insert a password. The responder did try the online update functionality but this did not work. Except one pacemaker programmer, all pacemaker programmers in this hospital are updated offline. The responder does not verify if the pacemaker programmer functions as expected after the update. However, in all the years they worked in this way, it never went wrong.

If a pacemaker programmer is broken or outdated the pacemaker programmer is collected by the vendor. However, the pacemaker, ICD's and other IMDs are collected by the funeral centre. Their destruction policy is not clear and for several doctors it was easy to get old IMDs for demonstration reasons.

Normally patients are not questioning pacemaker security. However, when there is a recent news item patients do. The responder mentions one case in which a shop security gate was wrongly programmed. Each time the patient walked into that store he felt sick.

# References


[1] *****. "**********". In: (2008).

[2] *****. "***** *****". In: (2010). URL: *****.

[3] *****. "*****". In: (2010). URL: *****.

[4] *****. "***** Programmer *****". In: (2011). URL: *****.

[5] R. Adams and C. Essex. *Calculus: A Complete Course*. Pearson Education Canada, 2009.

[6] H.Z. Al-Hassanieh, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science. *Encryption on the Air: Non-Invasive Security for Implantable Medical Devices*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2011. URL: http://books.google.nl/books?id=NuMYtwAACAAJ.

[7] R.L. Alexander. "Intelligent electronic device IED technology SCADA and 3 Oslash metering". In: (2002).

[8] R. J. Anderson. *Security engineering - a guide to building dependable distributed systems (2. ed.)* Wiley, 2008, pp. I–XL, 1–1040. ISBN: 978-0-470-06852-6.

[9] S. Azad, E. Altman and M. Haddad. "Jamming DoS in IEEE 802.11 WLANs". In: (2012).

[10] A. de Boer and J. Timmermans. "Blijvend in balans, een toekomstverkenning van informele zorg". In: (2007). URL: www.scp.nl/dsresource?objectid=19541&type=org.

[11] F.v.d. Broek. "Catching and Understanding GSM-Signales". In: *Radboud University Master Thesis* (2010).

[12] W. Burleson et al. "Design challenges for secure implantable medical devices". In: (2012), pp. 12 –17.

[13] W. Burleson et al. "Presentation Design Challenges for Secure Implantable Medical Devices". In: (2011). URL: http://www.cs.ru.nl/ifip-wg11.2/events/Slides-seminar2012/Burleson.pdf.

[14] A. Calder. *Information security based on ISO 27001 and ISO 17799- A Management Guide*. 2006.

[15] A. Calder. *Information Security based on ISO 27001 ISO 17799 A management Guide*. Van Haren Publishing, 2006.

[16] J. Care. "Secure Programming". In: (). URL: http://www.scribd.com/doc/2152150/Progamming-Security-Holes.

[17] J. A. Cazemier, P.L. Overbeek and L. M. C. Peters. *Best Practise for Security Management*. Office of Government Commerce, 1999.

[18] F.E. Cerullo. "Deploying Secure Web Applications with OWASP Resources". In: (2010).

[19] ETSI Technical Committee. In: (2009).

[20] M. Cova, C. Kruegel and G. Vigna. "Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code". In: (2010). URL: http://www.cs.ucsb.edu/~vigna/publications/2010_cova_kruegel_vigna_Wepawet.pdf.

[21] M. K. Daly. "The Advanced Persistent Threat (or Informa5onized Force Opera5ons)". In: (2009). URL: http://static.usenix.org/event/lisa09/tech/slides/daly.pdf.

[22] Deloitte. internal. 2013.

[23] T. Denning et al. "Patients, pacemakers, and implantable defibrillators: human values and security for wireless implantable medical devices". In: *Proceedings of the 28th international conference on Human factors in computing systems*. CHI '10. Atlanta, Georgia, USA: ACM, 2010, pp. 917–926. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753462. URL: http://doi.acm.org.janus.libr.tue.nl/10.1145/1753326.1753462.

[24] M. Dillinger, K. Madani and N. Alonistioti. *Software defined radio: architectures, systems, and functions*. ISBN: 0470851643.

[25] European Councel Directive. "90/385/EEC". In: (1990).

[26] I. Doorten. "De sociale dimensie van ouder worden". In: (2012). URL: http://www.rvz.net/uploads/docs/Achtergrondstudie_de_sociale_dimensie_van_ouder_worden.pdf.

[27] W.J. Elias. "The DBS system components: what lies beneath the skin". In: (2012). URL: http://people.virginia.edu/~rf3y/Elias/DBS_Devices.html.

[28] Eucomed. "Ultra Low Power Active Implants and Accessory Equipment". In: (2001). URL: http://www.eucomed.org/uploads/Modules/Publications/Ultra%20low%20power%20implants.pdf.

[29] M. Fähnle and M. Hauff. "analysis of unencrypted and encrypted wireless keyboard transmission implemented in GNU Radio based Software-Defined Radio". In: *Prof. Dr.Ing. Frowin Derr, Institute of Communication Technology* (2011).

[30] R. Fitzsimons. *Find a GSM base station manually using a USRP*. http://273k.net/gsm/find-a-gsm-base-station-manually-using-a-usrp/. 2007.

[31] M.J. Frank et al. "Hold Your Horses: Impulsivity, Deep Brain Stimulation, and Medication in Parkinsonism". In: *Science* 318 no 5854 pp 1309-1312 (2007).

[32] K. Fu. Interview. 2012. URL: http://www.wtop.com/807/2871848/Medical-implants-vulnerable-to-hackers.

[33] B. Furht et al. "Internet architectures for application service providers". In: *IEEE Internet Computing - INTERNET , vol. 2, no. 2, pp. 32-35* (1998).

[34] G. Garcia-Saez et al. "Architecture of a wireless Personal Assistant for telemedical diabetes care". In: *international journal of medical informatics 7 8 ( 2009 ) 391403* (2008).

[35] P. Gordon. "Data Leakage - Threats and Mitigation". In: (2007). URL: http://www.sans.org/reading_room/whitepapers/awareness/data-leakage-threats-mitigation_1931.

[36] P. Gould and A. Krahn. "Complications associated with implantable cardioverterdefibrillator replacement in response to device advisories." In: (2006).

[37] S. Gupta. "Implantable Medical Devices - Cyber Risks and Mitigation Approaches". In: (2012). URL: http://csrc.nist.gov/news_events/cps-workshop/cps-workshop_abstract-1_gupta.pdf.

[38] D. Halperin et al. "Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses". In: (2008), pp. 129 –142. ISSN: 1081-6011.

[39] D Halperin et al. "Security and Privacy for Implantable Medical Devices". In: *Pervasive Computing, IEEE* 7.1 (2008), pp. 30 –39. ISSN: 1536-1268.

[40] S. Hanna et al. "Take two software updates and see me in the morning: the case for software security evaluations of medical devices". In: *Proceedings of the 2nd USENIX conference on Health security and privacy*. HealthSec'11. San Francisco, CA: USENIX Association, 2011, pp. 6–6. URL: http://dl.acm.org/citation.cfm?id=2028026.2028032.

[41] G.M. Hardy. "Reducing Federal Systems Risk with the SANS 20 Critical Controls". In: (2012). URL: http://www.sans.org/reading_room/analysts_program/20CriticalControls.pdf.

[42] R. Hauser et al. "Deaths caused by the failure of Riata and Riata ST implantable cardioverter-defibrillator leads". In: 9.8 (2012). URL: http://download.journals.elsevierhealth.com/pdfs/journals/1547-5271/PIIS1547527112002913.pdf.

[43] X. Hei and X. Du. "Biometric-based two-level secure access control for Implantable Medical Devices during emergencies". In: (2011), pp. 346 –350.

[44] X. Hei et al. "Defending Resource Depletion Attacks on Implantable Medical Devices". In: (2010), pp. 1–5.

[45] A. Hildick-Smith. "Security for Critical Infrastructure SCADA Systems". In: (2005). URL: http://www.sans.org/reading_room/whitepapers/warfare/security-critical-infrastructure-scada-systems_1644.

[46] A. Hintz. "Fingerprinting websites using traffic analysis". In: (2002).

[47] J. Hoick. "4 perspectives on Web information systems". In: (2003).

[48] M. Howard, D. LeBlanc and J. Viega. *24 deadly sins of software security - programming flaws and how to fix them*. McGraw-Hill companies, 2010. ISBN: 978-0-07-162675-0.

[49] S. H. Huseby. *Innocent code - a security wake-up call for web programmers*. Wiley. ISBN: 0-470-85744-7.

[50] A.J. Jerri. "The Shannon sampling theorem Its various extensions and applications: A tutorial review". In: *Proceedings of the IEEE* 65.11 (1977), pp. 1565 –1596.

[51] M.E. Johnson and N.D. Willey. "Usability Failures and Healthcare Data Hemorrhages". In: *Security Privacy, IEEE* 9.2 (2011), pp. 35 –42.

[52] D Kang et al. "Analysis on cyber threats to SCADA systems". In: (2009), pp. 1 –4. DOI: 10.1109/TD-ASIA.2009.5357008.

[53] D. Kang et al. "Analysis on cyber threats to SCADA systems". In: *Transmission Distribution Conference Exposition: Asia and Pacific, 2009*. 2009, pp. 1 –4. DOI: 10.1109/TD-ASIA.2009.5357008.

[54] A Kerckhoffs. "La cryptographie militaire". In: *Journal des sciences militaires* (1883).

[55] E. D. Knapp. *Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress, 2011. ISBN: 1597496456.

[56] D. Kotz, S. Avancha and A. Baxi. "A privacy framework for mobile health and home-care systems". In: *Proceedings of the first ACM workshop on Security and privacy in medical and home-care systems*. SPIMACS '09. Chicago, Illinois, USA: ACM, 2009, pp. 1–12. ISBN: 978-1-60558-790-5. DOI: 10.1145/1655084.1655086. URL: http://doi.acm.org.janus.libr.tue.nl/10.1145/1655084.1655086.

[57] M.L. Kringelbach et al. "Translational principles of deep brain stimulation." In: *Nature reviews. Neuroscience* 8.8 (2007), pp. 623–635. URL: http://www.kringelbach.dk/papers/nrn_Kringelbach2007.pdf.

[58] U. Lakshmanadoss, P. Chinnachamy and J. P. Daubert. "Electromagnetic Interference of pacemakers". In: (2011).

[59] C. Langon. "All about modulation". In: (). URL: http://educypedia.karadimov.info/library/mod1.pdf.

[60] C. Langon. "Intuitive Guide to Principles of Communication". In: (). URL: http://www.complextoreal.com/chapters/fm.pdf.

[61] J.o Lee et al. "A 64 Channelprogrammable closed-loop deep brain stimulator with 8 channel neural amplifier and logarithmic ADC". In: (2008), pp. 76 –77.

[62] C. Li, A. Raghunathan and N.K. Jha. "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system". In: (2011), pp. 150 –156. DOI: 10.1109/HEALTH.2011.6026732.

[63] C. Liu et al. "Structural testing of Web applications". In: (2000).

[64] K.D. Mitnick and W.L. Simon. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, Inc., 2002. ISBN: 0471237124.

[65] R. Mojdehrakhsh et al. "Retrofitting software safety in an implantable medical device". In: *Software, IEEE* 11.1 (1994), pp. 41 –50. ISSN: 0740-7459. DOI: 10.1109/52.251205.

[66] Nasa. "Report Concerning Space Data System Standards, Telemetry: Summary of Concept and Rationale, Issue 1". In: (Dec. 1987).

[67] United States Government Accountabillity Office. "MEDICAL DEVICES FDA Should Expand Its Consideration of Information Security for Certain Types of Devices". In: (2012), p. 68.

[68] A. Okstra et al. "Vergrijzing en toekomstige ziektelastPrognose chronische ziektenprevalentie 2005-2025." In: *RIVM - Bilthoven - the Netherlands* (2007), p. 89.

[69] D. Panescu. "Emerging Technologies [wireless communication systems for implantable medical devices]". In: *Engineering in Medicine and Biology Magazine, IEEE* 27.2 (2008), pp. 96 –101.

[70] P. W. Parfomak. "Pipeline Cybersecurity: Federal Policy". In: *Congressional Research Service* R42660 (). URL: http://www.crs.gov.

[71] n. Paul, T. Kohno and DC. Klonoff. "A review of the security of insulin pump infusion systems." In: (2011).

[72] S. N. Premnath and S. K. Kasera. "Battery Draining Denial of Service Attack on Bluetooth Devices". In: (2008). URL: http://www.cs.utah.edu/~nandha/Abstract_2008.pdf.

[73] J. Scambray, S. McClure and G. Kurtz. *Hacking exposed second edition*. Academic Service, 2000.

[74] B. Schoenmakers. "Lecture notes Cryptographic Protocols". In: (2011).

[75] M.I. Sedney et al. "Hergebruik van pacemakers". In: (1986). URL: http://www.ntvg.nl/publicatie/Hergebruik-van-pacemakers/volledig/print.

[76] A. Silberschatz, H.F. Korth and S. Sudarshan. *Database system concepts*. McGraw-Hill, 2011. ISBN: 978-007-128959-7.

[77] M. Silic, J. Krolo and G. Delac. "Security vulnerabilities in modern web browser architecture". In: (2010), pp. 1240 –1245.

[78] I. Sinclair and J. Dunton. *Electronic and Electrical Serving — consumer and commercial Electronics level 3*. 2002. ISBN: 1136407456.

[79] G. Smithson. "Introduction to Digital modulation schemes". In: *Plextek Ltd Magazine, Great Chesterford, pp. 1-9* (). URL: http://educypedia.karadimov.info/library/schmsv6.pdf.

[80] F. Stajano and R.J. Anderson. "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks". In: *Proceedings of the 7th International Workshop on Security Protocols*. London, UK, UK: Springer-Verlag, 2000, pp. 172–194. ISBN: 3-540-67381-4. URL: http://dl.acm.org/citation.cfm?id=647217.760118.

[81] S. Stanslaski et al. "An implantable Bi-directional brain-machine interface system for chronic neuroprosthesis research". In: (2009).

[82] Parkinson's support and research charity. "Deep brain stimulation - Parkinson's surgery". In: (2012). URL: http://www.parkinsons.org.uk/about_parkinsons/treating_parkinsons/surgery/deep_brain_stimulation.aspx.

[83] P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005. ISBN: 0321304543.

[84] Y. Tadaaki. "Measures against police information leakage utilizing automatic encryption and access control software". In: (2008).

[85] H. C.A. V. Tilborg. *Fundamentals of Cryptology*. Kluwer Academic Publishers, 1999. ISBN: 0792386752.

[86] R. Tsang. "Cyberthreats, Vulnerabilities and Attacks on SCADA Networks". In: (2010).

[87] K. Tsipenyuk. "Seven pernicious kingdoms: A taxonomy of software security errors". In: (2005), pp. 36–43.

[88] Department of Defence United States. "Supervisory control and data acquisition (scada) systems for command, control, communications, computer, intelligence, surveillance and reconnaissance (c4isr) facilities". In: *Headquarters Department of the army Washington, DC* (2006).

[89] R. Vidgen et al. *Developing web information systems*. Elsevier, 2002. ISBN: 978-0-7506-5763-1.

[90] J.A. Warren et al. "Implantable Cardioverter Defibrillators". In: *Proceedings of the IEEE* (1996).

[91] J.G. Webster. "Implantable Cardioverter Defibrillators". In: (2004).

[92] D. Wessels. "Implantable pacemakers and defibrillators: device overview amp; EMI considerations". In: *Electromagnetic Compatibility, 2002. EMC 2002. IEEE International Symposium on*. Vol. 2. 2002, 911 –915 vol.2. DOI: 10.1109/ISEMC.2002.1032815.

[93] D. Wichers. "Getting Started with OWASP The Top 10, ASVS, and the Guides". In: (2010). URL: https://buildsecurityin.us-cert.gov/swa/presentations_2010_10/01_Monday/OWASP_Presenters/03_Dave_Wichers_2010-DC_SW_Assurance_OWASP_Projectsx.pdf.

[94] K.A. Wolnik et al. "The Tylenol tampering incident tracing the source". In: 56 (1984).

[95] J. Wyffels et al. "A novel indoor localization system for healthcare environments". In: (2012), pp. 1 –6.

[96] Y. Zhang, P.L. Jones and R. Jetley. "A Hazard Analysis for a Generic Insulin Infusion Pump". In: *Journal of Diabetes Science and Technology Volume 4, Issue 2, March 2010* (2010).

[97] P. Zhende. "Survey on Vulnerabilities, Threats and Security Methods of DBMS". In: *Proceedings of the NCNTE-2012, Third Biennial National Conference on Nascent Technologies* (2012).

[98] B. Zhu, A. Joseph and S. Sastry. "A Taxonomy of Cyber Attacks on SCADA Systems". In: *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*. 2011, pp. 380 –388. DOI: 10.1109/iThings/CPSCom.2011.34.