

Experimental Security Analysis of Connected Pacemakers

Guillaume Bour¹^a, Marie Elisabeth Gaup Moe²^b and Ravishankar Borgaonkar¹^c

¹*SINTEF Digital, Strindvegen 4, Trondheim, Norway*

²*Department of Information Security and Communication Technology, NTNU - Norwegian University of Science and Technology, Trondheim, Norway*
guillaume.bour@sintef.no, marie.moe@ntnu.no, ravi.borgaonkar@ieee.org

Keywords: Pacemaker, Connected IMD, Medical Device, Cyber Security, Home Monitoring Device Security

Abstract: Medical devices and their connectivity capabilities are providing a variety of benefits to the healthcare domain, including remote monitoring, automated alerts, and improved patient outcomes. However, these medical devices introduce a range of new potential cyber security risks when connected to the Internet, affecting the patient or the healthcare infrastructure. In this paper, we systematically analyze the security issues of connected pacemakers. In particular, we use a black box testing methodology against a commercial pacemaker device and the network infrastructure. Our main objective is to understand how the data is sent from a bedside monitor in the patient's home to the backend server hosted by the pacemaker manufacturer, and whether or not this data is protected from a cyber security perspective. To do so, we leveraged several hardware related vulnerabilities found in the bedside monitor to obtain the firmware of the device and then reverse engineered the proprietary communication protocol. We demonstrate how vulnerabilities in this protocol can be leveraged to allow an attacker to perform a man-in-the-middle attack on the pacemaker.

1 Introduction


Implantable Medical Devices (IMD) in the form of modern pacemakers are not a new medical innovation but the evolution of technology from the fifties and sixties. In the seventies “on-demand” pacemakers were developed that would sense the patient's cardiac activity and adjust the pacing to it. These pacemakers could be remotely programmed through a radio-frequency telemetry link. The first pacemakers driven by microprocessors appeared in the nineties. These devices were able to detect cardiac events and could adapt their internal pacing based on the patient's needs. The first *connected* pacemakers appeared in the early 2000s, with the addition of an external device that would connect wirelessly to the pacemaker and upload its data to a remote server via the Internet, thus reducing the need for patients to go to the clinic for a check-up. Today, this remote connectivity is getting more and more popular in use. An external device, sometimes called a “bedside monitor”, which we in this paper will refer to as the Home Monitoring Unit (HMU), is used to gather the pace-


maker's data and upload it to a remote server accessible to the clinician through a web interface.


Patients' safety has always been a key priority in medical devices. Pacemakers are built with “fail-safe” modes which they will switch to in case something goes wrong with its programming, to keep the pacemaker generating a constant pulse until the patient gets seen by a pacemaker technician that can re-program the device.

Cyber security of medical devices, on the other hand, has not been paid much attention by researchers or publicly debated until the last decade. The healthcare domain is however not spared by cyber criminals, and attacks like the WannaCry ransomware that struck the world in May 2017 has shown that hospitals and medical devices are at risk for being infected via collateral damage even if the attack was not specifically targeted towards them, and that a cyber attack can have a real impact on human lives.

Connected devices in the form of wearables like the Apple Watch or the Fitbit have become increasingly popular, and by extension, we are getting used to having access to close to live data on our health. This trend highlights the need for data protection for all the devices that monitor our health, including IMDs. Medical data is indeed an interesting target for

^a  <https://orcid.org/0000-0003-4456-6279>

^b  <https://orcid.org/0000-0003-1786-1133>

^c  <https://orcid.org/0000-0003-2874-3650>

criminals, who can monetize it by selling it to other criminals on underground forums. As detailed in an article by Robbie Richards [13], “*Criminal attacks are now the number one cause of security breaches in healthcare, increasing 125% since 2010.*” With medical devices being more and more connected, the attack surface is growing, and it is thus important to design the devices with security in mind and not to rely on “security by obscurity” or “bolt-on security”, which tends to be often the case when security comes as an afterthought.

Over the past three years, we have been analyzing the security of the pacemaker ecosystem of one of the main vendors on the market today. We looked at three different generations of HMU devices and compared their security to document the state-of-the-art and to see how security implementation in these devices evolved over time.

Our results suggest that, even if the overall security of the devices has improved, the medical device manufacturers are still lagging behind and fail to implement some common security practices.

The paper is organized as follows. Section 2 provides the background of our work, including a description of the principle of the pacemaker and its ecosystem, along with the interactions between its different components. Here we also review the relevant related work and the threat model used in our research. In Section 3 we outline the methodology used along with the setup used to perform the security analysis. Section 4 presents our main findings, from a hardware, network, and devices management perspective. Section 5 provides a discussion of the results along with mitigations. Section 6 concludes the paper.

2 Background

2.1 The Pacemaker Ecosystem

Pacemakers and Implantable Cardioverter Defibrillator (ICD) are active implantable medical devices, which are defined in the Norwegian regulatory framework [8] as “*Any 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.*” Both pacemakers and ICDs are battery-powered devices surgically implanted in a patient to treat a heart related condition. They differ in the conditions they are treating, as ICDs are not only capable of continuous monitoring the heart rhythm and pacing the heart with elec-

trical pulses, but also to deliver an electrical shock to the heart if required. In this paper, because both ICDs and pacemakers are similar devices from the cyber security point of view, they will both be referred to as pacemakers.

Pacemakers are constructed to last for around 10 years varying on their usage, before having to be replaced due to the battery running out. The devices are able to deliver pacing when required and in a way that is adapted to each patient. This means that the clinician needs a way to program the device in a non-invasive way for the patient. As previously mentioned, a RF-telemetry link was introduced to the devices in the 70s to program some parameters in the pacemaker. Since then, pacemakers have evolved into complex embedded devices, driven by a microcontroller. Once implanted, they are not standalone devices which are left there for ten years waiting to be replaced, but take place in an ecosystem that allows for monitoring the devices and also the patients’ condition.

The Home Monitoring Unit is an example of a communication device that enables this remote regular monitoring by connecting wirelessly to the pacemaker, reading and transmitting data from it. Figure 1 presents the pacemaker ecosystem.

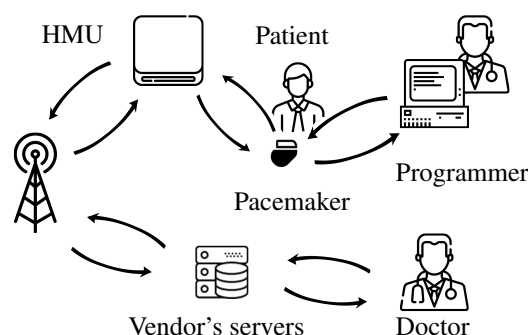


Figure 1: Diagram of the vendor’s pacemaker ecosystem. [2]

The pacemaker Implanted in the patient’s body, this is the main device of the ecosystem. As already explained, it generates an electric impulse that helps regulate the heart rhythm.

The programmer The programmer is an external computer used by a clinician to program the pacemaker. Programming the pacemaker is achieved wirelessly by placing the programming head of the programmer in close proximity of the pacemaker. While old pacemakers used to communicate with the programmer over the 175 kHz band, newer ones tend to use 402-405 MHz Medical Implant Communications (MICS) band [16]. The communication of the pace-

maker with the programmer is triggered by applying a magnetic field on the implant, causing a magnetic switch inside it to close [6]. This magnetic field is emitted by the programming head. It is to be noted that pacemakers from different vendors require different programmers due to differences in communication protocols, and that a programmer of a specific vendor usually supports several pacemakers/ICD devices from the same vendor.

The Home Monitoring Unit The HMU is a router-like device in charge of collecting telemetry data from the implant and transmit it. The device is paired with a pacemaker, placed in the patient's home and receives the data sent by the pacemaker at a pre-configured time (for instance every night at 2:00). The HMU also communicates with the pacemaker over the 402-405 MICS band, which allows for longer range communications than the 175 kHz band. This data is then sent to a backend server, usually owned by the pacemaker manufacturers. Similarly to the programmer, pacemakers from different manufacturers requires different HMUs. Some newer pacemakers communicate over Bluetooth Low Energy with an app installed on the patient's smartphone, eliminating the need for an external HMU device.

The operator's network In order to transmit the data to the backend server, the HMU needs connectivity. To achieve that without having to rely on patients' internet connection and also for ease of use, manufacturers usually have contracts with Telecom operators. That way, the HMUs are shipped with a SIM card to access the GSM or 3G networks, or with access to the internet through telephone lines for older versions. The HMU either connects directly to the server which is exposed on the public Internet or connects to a Virtual Private Mobile Network (VPMN) which gives it access to the server. This implementation varies with vendors.

Vendor's backend infrastructure This infrastructure is used to receive data sent by the HMU, process it and make it accessible to the clinician through an online platform. Alerts may also be triggered if something looks irregular, for instance, if no data has been uploaded in a while for a given HMU, or if there is a problem with the patient's condition. This allows the clinician to call in the patient for a follow-up checkup if necessary.

2.2 Pacemaker Security Related Research

While wireless communication technology has been a feature of pacemakers since the seventies, security

researchers have only been taking an interest in this topic for around 15 years. In 2008, Halperin et al. published the first research paper describing a security attack against a commercial pacemaker [6]. Their research targeted the communication between the pacemaker and its programmer. Using Software-Defined-Radio (SDR), they partially reverse engineered the communication protocol in use and, with that knowledge, were able not only to eavesdrop and decode the communication, but also to perform data replay attacks. They were able to interrogate the pacemaker to reveal the patient's data containing personal information such as patient's name, diagnosis, etc. They were also able to change parameters of the pacemaker, such as the patient's name, implantation date, or even therapies (that includes turning off all therapies). Finally, and more frightening, they were able to trigger a shock on the ICD, which could have fatal consequences on a real patient if delivered at an inappropriate time. They thus highlighted the severity of the lack of security mechanisms for implantable medical devices.

In more recent research from 2016, Marin et al. carried out similar research on the latest generation of pacemakers [10]. Their research highlights several weaknesses in the communication protocol and shows that a weak adversary can perform attacks even with low capabilities. Three kinds of attacks were performed. First, the researchers managed to access private patient information from the telemetry information, even though some obfuscation technique was done by the manufacturer. Secondly, they performed Denial-of-Service attacks. By keeping the device in "interrogation" mode, they were able to send messages to the device over a long-range communication channel and thus drastically reduce the implant battery life. Finally, they found that there is no mechanism against replay attacks and that an adversary without any knowledge of the protocol could simply replay captured messages and spoof the programmer.

A report exposing vulnerabilities in the pacemakers and HMUs manufactured by St. Jude (now Abbott) was published by Muddy Waters Capital LLC in 2016 [1]. Amongst the vulnerabilities that were presented was a way to perform a battery-draining attack on the pacemakers or forcing them to pace at a rhythm that would be potentially fatal for the patient. These attacks were carried out by first compromising the HMU, which was then used to attack the pacemaker. Even if no attack has been publicly reported exploiting these vulnerabilities, the disclosure of this report had a potentially severe impact on the 260 000 HMUs deployed in patients' homes at the time. As a result, the vendor issued a firmware update at the

beginning of 2017 to mitigate the vulnerabilities.

In 2017, Rios and Butts evaluated the security of the pacemaker ecosystems of the four major vendors [14]. They presented several weaknesses, in the programmers, the pacemaker implants, and the HMUs. Weaknesses include vulnerable third-party software, lack of authentication between devices, unencrypted filesystems and firmware, removable hard-drives, and unsigned firmware. The conclusion is that the whole industry is quite immature in terms of cyber security. They highlight that this is not only the case for one unique vendor but that all vendors are impacted.

2.3 Threat Model

In this paper, we aim at understanding the evolution of the security measures in the pacemaker ecosystem and to evaluate its current maturity. In our research, we have considered two classes of adversaries:

With physical access to an HMU It is possible to buy these devices online, sometimes at the low price range of \$20 - \$50. Since these are much easier to obtain (compared to a programmer), one can afford to experiment with them without the fear of breaking an expensive device.

Capable of setting up a Fake Base Station (FBS) Such an attacker has access to Software Defined Radio equipment, which is also affordable. The HackRF One¹ manufactured by Great Scott Gadgets costs around \$350.

The two main assets that we want to look at in this research are the patient's safety and privacy. As a consequence, we study the impact of different attacks on the patient's treatment and how it could be interfered with, directly or indirectly. Regarding the patient's privacy, we look at what attacks would enable an attacker to access any kind of private data about the patient.

As mentioned in the introduction of this paper, motivations to attack the pacemaker ecosystem vary. Attacks against the patients' privacy are mostly driven by financial motives, in order to monetize the medical data on the black market. These attacks can have a big impact if they can be leveraged at a large scale. Safety related attacks could also be motivated by financial profit, for example we can imagine that an adversary could leverage a vulnerability in an extortion attempt by threatening a patient or maybe even a medical device manufacturer asking for a ransom. Targeted attacks against a single individual in order to harm or kill are less likely, except if it is a person

¹See <https://greatscottgadgets.com/hackrf/>

of high interest. In both cases, one can imagine that we are facing organized crime or a nation-state threat actor. However, we cannot exclude single opportunistic attackers.

3 Methodology and experimental environment

3.1 Black Box Testing Methodology and Target

Our research focused on the Home Monitoring Unit, more specifically, two main attack vectors were studied: *physical* and *network*. In both cases, a black box testing methodology was followed, as the tested components were proprietary hardware and software of which we had very little knowledge. In order to be as close to a real-world scenario as possible, we used commercial off-the-shelf (COTS) equipment whenever possible, and tried to keep the cost of an attack as low as possible.

The targeted devices from our lab were acquired second hand and are all BIOTRONIK's devices. This manufacturer was chosen because no prior security research had been published for this particular pacemaker brand, and no known vulnerabilities had previously been disclosed for its devices. Devices in our research project include three different generations of the HMU:

- V1** : From the early 2000, one of the first HMUs on the market, using the GSM network for connectivity. This version is not used anymore.
- V2** : From the 2010s, in two models: one using the plain old telephone service for its connectivity, the other one using the GSM network. This version is not commercially available anymore, but is still in use.
- V3** : From 2016, using the 3G network. This is the current commercially available version.

3.2 Testing Setups

To perform hardware testing, as previously mentioned, we used a combination of several COTS equipment: *The shikra* to interface with low-level data interfaces via USB, a *logic analyzer* (useful to detect non standard baud rate), the *JTAGulator* to identify the pinout of the JTAG interface (it does so by trying all possible permutations). Once the pins were known, we used a *Raspberry Pi zero* as our UART/JTAG/SPI connector.

From a network perspective, we used two different setups:

- We developed a *modem emulator* to interact with the Telephone Line version of the HMU (V2).
- A *Fake Base Station* (based on OpenBTS) to interact with the GSM (V2) and 3G (V3) versions of the HMUs. In addition, a network jammer was used to prevent the HMU from connecting via the 3G network, forcing a so-called downgrade attack. A virtual machine was set up to emulate the backend server of the manufacturer (as shown in Figure 2).

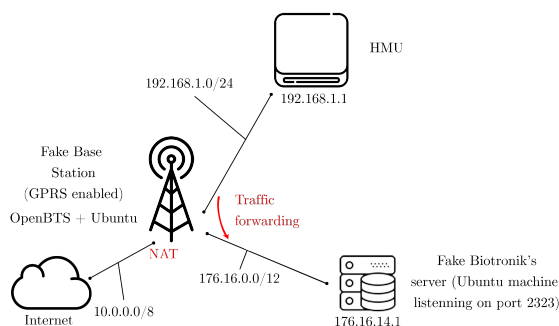


Figure 2: Network diagram of the emulated network

3.3 Ethical considerations

Given that the devices available in our lab have been acquired on the second-hand market, and that some of them were not new, they could have contained potentially sensitive data. This data has been systematically redacted from this paper and from previous publications. The Norwegian Centre for Research Data (NSD) was notified at the beginning of our project, and approved our patient data protection plan.

As the vulnerabilities discovered in the pacemaker ecosystem during our research could have had a potential impact on patients' safety and security, our findings were kept under embargo for one year. During this time the research findings were shared with the vendor (BIOTRONIK) in the form of a vulnerability report. The vendor cooperated according to a coordinated vulnerability disclosure process and appropriately analyzed and validated our report. They then shared their responses to each reported vulnerability, and we discussed each point in detail. During these discussions, they also provided sufficient information to confirm that patient harm arising from the vulnerabilities is very unlikely. BIOTRONIK recommends that healthcare providers and patients continue to use the investigated devices as intended and

follow device labelling. The coordinated vulnerability disclosure process also involved the German Federal Office for Information Security (BSI), the German Federal Institute for Drugs and Medical Devices (BfArM), the US Cyber Security and Infrastructure Security Agency (CISA) and the US Food and Drug Administration (FDA). As a result of this process, CISA issued an advisory [4].

4 Security Analysis of the HMU

Our main objective with this project was to better understand how the data is sent from the HMU to the vendor's backend servers, what kind of vulnerabilities exist on this part of the ecosystem and how they could be exploited to potentially impact patient's safety and/or privacy. In the following sections, we explore two classes of attack: the first focuses on exploiting hardware vulnerabilities to compromise a Home Monitoring Unit, and the second on how an attacker can exploit vulnerabilities on the home monitoring unit to gain access to the backend infrastructure.

4.1 Hardware security analysis

During the hardware analysis, we found several vulnerabilities:

Debug interfaces available. On all HMU versions analyzed, we were able to discover the UART and JTAG interfaces. On versions 1 and 2 the pins were not labelled, making it harder to determine the JTAG interface. On the latest version, however, pins were labelled. On all versions, the UART interface seemed disabled, and it was not possible to interact with it. The JTAG interface on the other hand was enabled and it is possible to fully control the microcontroller using it. That includes dumping the contents of the Random Access Memory (RAM) as well as the Flash Memory, which gave access to the firmware of the device.

Credentials are sent in clear text to the modem.

When analyzing the version 2 of the HMU, we were able to eavesdrop the communication between the microcontroller and the modem as the pins of the modem were exposed on the PCB. This allowed us to get access to the credentials used by the device to connect to the manufacturer's Virtual Private Mobile Network (VPMN), since these were sent in clear text.

Firmware is not encrypted, nor obfuscated. Once the firmware was dumped via the JTAG interface,

reverse-engineering revealed that it was not encrypted or protected in any way. There was no trace of obfuscation of the code. On the contrary, log strings used by the device were explicit enough to ease the process of reverse engineering. This made it possible to create a script to easily fetch the credentials previously acquired via eavesdropping on the communication channel directly from the firmware, along with other credentials used by the device to connect to the backend server hosted by the manufacturer.

Memory is not encrypted. The memory is not encrypted either, meaning that anyone with physical access to the HMU can copy it via the JTAG interface and access the data going through the HMU, including the patient’s data.

Hard-coded credentials and cryptographic keys. The credentials used by the devices to connect to the network and backend servers are hard-coded and stays the same for each connection attempt, we observed however that they are unique for every device (two different HMUs will use different credentials). On the latest version they are stored on the external flash which is not encrypted and whose content can be read via the Serial Peripheral Interface (SPI). Cryptographic material such as DES and AES keys used in the proprietary protocol is also stored in a similar way.

Unencrypted communication with the pacemaker Even though we have not done exhaustive research at this interface due to limited access to working compatible pacemakers in our lab, we found that there is no encryption of the data exchanged between the pacemaker and the HMU. That means that attackers who can intercept the radio signal from the pacemaker (the radio band is already known) can also access the patient’s data.

Leveraging the hardware vulnerabilities exposed above and the raw network data obtained through the modem and server spoofing, we were able to reverse engineer the proprietary communication protocol used by the HMU to send the pacemaker data to the backend server. The detailed structure of a data packet is presented in Figure 3.

4.2 Network security analysis

When analyzing the security of the communication link between the HMU and the backend server, we identified several weaknesses in the communication protocol.

No mutual authentication. We were able to spoof the backend server and trick the HMU into sending

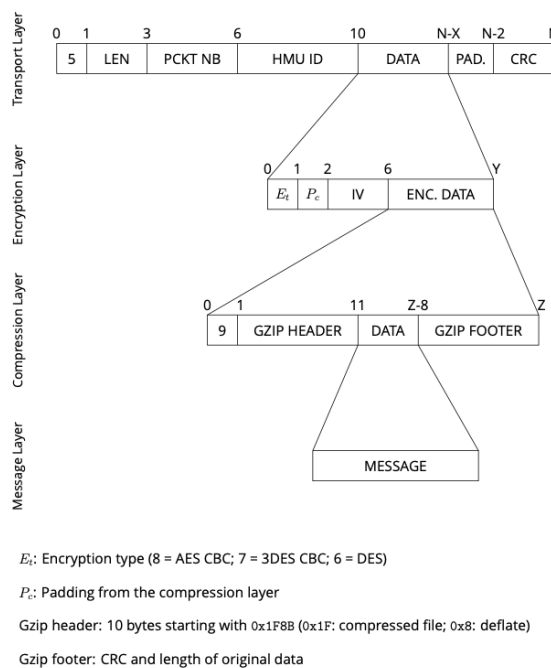


Figure 3: Structure of the communication protocol’s packet

its data to us, highlighting the lack of mutual authentication between the backend server and the HMU on the first two versions. We did so at two different levels: first at the modem level on the Telephone line version, where we spoofed both the modem and the backend server, and at the network level, where we used a virtual machine, connected to the same network as our OpenBTS machine, with the proper IP address requested by the HMU, to respond to the TCP request of the HMU (see Figure 2). The data obtained was encrypted for the most part. However, credentials to connect to the service were sent in cleartext before switching to the encrypted communication.

Usage of a proprietary protocol over an insecure transport protocol.

The version 2 and 3 of the HMU use both GPRS and SMS to send data. On both channels, the data is sent using the proprietary protocol presented in section 4.1 on top of TCP. This protocol packs, compresses (when using GPRS) and encrypts the data.

Broken or risky algorithm. In the case of patient data, the proprietary protocol uses AES CBC as the encryption algorithm, however single DES is used in the case of log data going over SMS. DES is a broken algorithm from a security perspective, and log data can thus easily be obtained by an attacker that is able to set up a Fake Base Station in the proximity of the HMU. An attacker having had physical access once to the HMU can also perform the same attack on patient data by getting hold of the AES key. The keys

(AES and DES) were however random and unique per device.

Credentials reuse. The credentials used to connect the VPMN and the backend services are the same and are sent unencrypted in both cases. They are thus very easy to obtain.

By chaining several of the vulnerabilities, we were able to *weaponize* the second version of the HMU. With physical access to the device, an attacker can install a physical device with a wireless communication interface inside of it (the inside of the HMU casing is big enough to add a *RaspberryPi zero*), and that way gain *remote access* to the device. This allows an attacker to not only eavesdrop on all communications between the HMU and the backend server, but also to act as a *Man-in-the-Middle*, the proprietary protocol being known. Such an adversary can also get access to all the data sent by the pacemaker to the HMU. This would enable an attacker to modify the pacemaker telemetry data in order to hide a possible problem, or to create a problem by deleting or modifying pacemaker alerts and warnings that were meant to be sent to the backend server.

4.3 Credentials & SIM cards validity

The HMU has two sets of credentials: the first to connect to the network and access the manufacturer VPMN; the second to connect to the service on the backend server. To verify the validity of the credentials, we used them on a phone with the HMU SIM card and manually entered the settings in order to connect to the VPMN. However, when using the version 2 HMU’s SIM card, we were unable to connect because the SIM card was not valid anymore.

It turned out that using a SIM card from an old first version HMU on a newer second version HMU worked: we were able to connect to the VPMN and obtain an IP address inside the VPMN. To ensure we were in the right network, a successful ping request was sent to the server hosting the telemetry collection service. No other testing was performed as this was outside of our research scope and could potentially interfere with the manufacturer’s service.

The VPMN is an additional security measure, even if this is not its main purpose. It prevents the patient data telemetry servers from being publicly exposed to the Internet, something that for instance protects against Denial-of-Service attacks. However, as shown in our research this protection can be bypassed by an attacker who acquires an old device with a valid SIM card, highlighting the need for proper decommissioning procedures for old devices.

4.4 Attack scenarios

In this subsection, we will describe two attack scenarios against the HMU and more generally against the whole pacemaker ecosystem. Figure 4 and 5 present the attack trees for these scenarios. Arrows indicate a requirement. An arc between several arrows indicates an “AND” condition while single arrows indicate an “OR” condition.

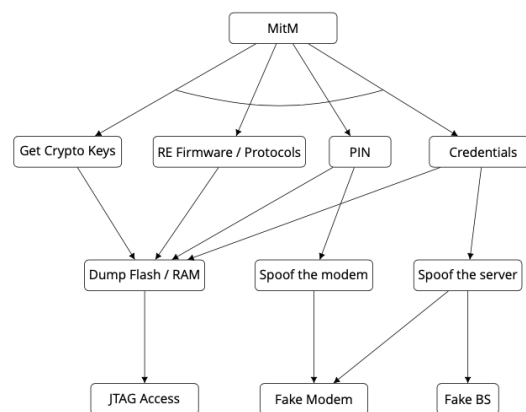


Figure 4: Attack tree for the “MitM” scenario

The first scenario is the *Man in the Middle* presented in Figure 4. Given the vulnerabilities described earlier, an attacker can spoof the identity of the HMU for the backend server and vice-versa. This means that an adversary can have full control over the information that is sent between these two entities. In order to target a patient an attacker could for instance constantly send good reports, suppressing any alerts or warnings from the pacemaker. This could trick a clinician into thinking that the patient is doing great while in reality, the patient might be in urgent need of a check-up, for example, due to the pacemaker battery running out. Having an HMU would thus be more dangerous than having no home monitoring enabled, due to a false sense of security and potentially fewer clinic visits.

The second scenario can be described as *Unauthorized access to the backend server* and is presented in Figure 5. We believe that this is possible with both versions of the HMU, given that the attacker can access credentials that are still valid. The attack tree is only showing the GSM attack tree, the attack tree for the T-Line would be similar but easier since it only requires a working telephone line and no valid PIN or SIM. If an attacker can access the Virtual Private Network (VPN) with their computer using the credentials of the HMU, they would have direct access to the backend server (and all machines that reside in the same private network unless proper network seg-

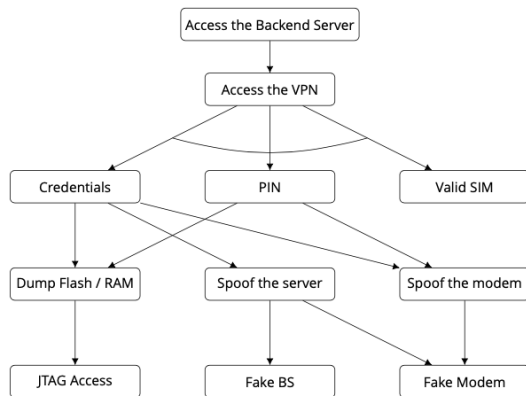


Figure 5: Attack tree for the “Unauthorized access to the backend server” scenario

mentation with security monitoring is in place). If any of these machines are compromised, the result could be a significant data leak of personal data. Second-hand HMUs can be bought for a very low price on the internet, some come with their SIM cards still valid as we have demonstrated in our research, thus enabling an attacker to perform such an attack.

5 Discussion

5.1 Results from the security analysis

Our research confirms what was highlighted by Rios and Butts: the industry is overall still quite immature when it comes to cybersecurity [14]. Indeed, from a hardware point of view, an attacker with physical access to a device can easily get access to patients’ data with no need for extensive knowledge or expensive equipment. From our observations, best security practices were not applied when it comes to hardware security given our findings of vulnerabilities that can all be described as commonly found in embedded devices. From a network perspective, several weaknesses have been identified in the protocol that is used by the HMU to communicate with the backend server, such as the credentials sent in clear text over TCP, the usage of a weak cipher to send data using SMS or the lack of mutual authentication in the second version of the HMU.

On balance, it is also important to highlight that there is a notable evolution in terms of security between the versions. The latest version of the HMU seems to implement mutual authentication and stronger cryptographic ciphers than the previous versions. We can also point out that even though the second version has been found to have several weaknesses and vulnerabilities, the telemetry data was al-

ready encrypted using AES CBC with the keys being randomly generated and unique per device.

5.2 On the trade-offs in the medical industry

When designing IMDs, there are several security objectives to have in mind. These are the regular six following properties: Confidentiality, Integrity, Availability, Non-repudiation, Authorization and Authentication. One also needs to consider the two modes under which these security properties have to be respected: *normal operation mode* and *emergency mode*. In the normal operation mode, the patient is in a state where it is reasonable to assume strict control of which devices can interact with the IMD, and it is feasible to implement strong access control, through the mean of cryptographic protocols for instance. Camara et al. explain that ideally, the device should not be detectable by unauthorized parties in this mode, and should “*ignore data requests or device reprogramming commands*” [3]. In emergency mode, even though the previously mentioned security objectives are important, it is vital that the device be accessible, for example if the patient is to undergo an emergency procedure for which the pacemaker must be deactivated.

It is thus a challenge for manufacturers to develop devices that fulfill all these characteristics. Zheng et al. highlight the trade-offs that come with the pacemaker ecosystem [20]. The first one is related to the emergency mode: *security vs accessibility*. Indeed, the pacemaker purpose is to save the patient’s life and should not be an obstacle during an emergency surgery. The second trade-off is, *emergency access vs secure checkup access*. Securing the regular access while having an emergency access, which is almost like a security backdoor, is a challenge, especially when one must also take into account the battery life of the device. This leads to the third trade-off which is *limited resources vs strong cryptography*. Indeed, to secure the device, one need to implement strong cryptography which require intense processing power, this conflicts with the with low power capabilities and the long battery life time required by the implanted devices. This can even be abused by an attacker that launches Denial of Service (DoS) attacks in the form of constant wireless communication requests to draw the device battery, leading to a premature pacemaker battery depletion which requires surgery and thus setting the patient’s life at higher risk from complications.

5.3 Mitigations & defense mechanisms

As mentioned in the previous section, building safe and secure medical devices means facing several trade-offs. Several solutions have been proposed to solve the problem of having a secure access to the device while allowing access in emergency situations. Zheng et al. wrote a review [18] of the different mechanisms that could be used:

External proxy-based solutions. This idea was first proposed by Denning et al., and consists, as the name indicates, in having an external device called the Communication Cloakers to protect the implant [5]. This external device is carried by the patient and protects the implant from attacks in the everyday life. In an emergency situation, when the clinician doesn't have access to the distributed key, they can simply remove the proxy. However, this also means that if patients forget or lose their proxy device, their implant becomes vulnerable to attacks again.

Biometric-based access control. This type of solutions uses patients' biometric features in order to provide access. For instance, the Heart-to-Heart (H2H) scheme [15] makes sure that the pacemaker can only be accessed by a programmer in physical contact with the patient by using ECG signals to generate the crypto material to establish a secure wireless communication. Other solutions might use different biometric features, such as fingerprints, iris or even voice [19].

Proximity-based security schemes. In these schemes, the proximity of the device is used to determine whether or not a functionality is available. For instance, changing the device settings, which is a critical operation, requires close proximity (a few centimeters) while home monitoring is allowed up to 10m. If some authentication scheme such as Ultrasonic-AC [12] combine proximity and security credentials, others can be based on magnetic fields or short-range communication protocol. This is in fact what is currently used to secure pacemakers. It has however been proven to be vulnerable if the attacker uses strong magnetic fields or simply use powerful and sensitive transceivers and high gain antennas [10].

Key distribution supporting emergency access. These schemes rely on cryptography to achieve secure access in normal situations while also keeping an emergency access. This includes symmetric cryptography, in which the key is distributed to authorized devices. For emergency situations, it is proposed that the key is carried by the patient, either with a smart card, on a bracelet or simply tattooed on

the skin (with UV ink for instance) [17]. Public key cryptography can also be used. However, in the case of an emergency situation, the programmer needs to contact a trusted party to obtain the certificate that can be used to derive a symmetric key, and this requires access to the Internet. In addition, public key cryptography is not compatible with the low energy requirements of the pacemakers. Finally, it is possible to use biometric features to generate keys, as already explained for biometric-based solutions.

In their review, Zheng et al. also suggest possible solutions to address the resource constraints of IMDs [18].

Lightweight cryptography. In order to preserve the implant energy, manufacturers need to use lightweight cryptography protocols. Marin et al. propose a key agreement protocol that is an alternative to the proposal of Halperin et al., that was to add a standard symmetric key authentication and encryption between the ICD and the programmer, thus requiring the key to be safely stored on the programmer and opening the door to it being leaked. Marin et al. propose a semi-offline protocol: the IMD is in charge of computing a new key for the new period. To do so, the programmers need to contact the vendor to obtain the key for the period. That way, if the programmer is lost, or not in use anymore, it will not receive any updated key, and thus the ecosystem goes back to a secure state when the key is changed. Even though the new key computation is expensive for the IMD, this is a rare event and is thus not a problem.

Energy Harvesting. Another way to protect medical devices is through energy harvesting. Halperin et al. propose zero-power defenses for IMDs. These defenses includes detection of attacks, prevention of attacks and a key exchange mechanism. As a detection mechanism, they propose to add a way to make the patient aware that there is something out of the ordinary happening, by for instance playing a "beep" if the security is disabled on the implant. The zero-power idea is to use a piezo-element driven by wireless communication (thus alerting a patient that wireless communication is happening).

Separate security unit. Last but not least, the usage of a separate security unit that would be in charge of the security can mitigate the impact on the battery of the implant. This is for instance something that can be pushed to the external proxy device proposed above.

Moving away from the communication channel, another area that needs improvements is the software security. Li et al. propose a way to improve the

trustworthiness of medical device software with formal verification methods [9]. They applied their approach to the firmware of a pacemaker and “demonstrated its ability to detect a range of software vulnerabilities that compromise security and safety.” The idea behind formal verification is not only to check for common vulnerabilities such as buffer overflows, use after free, etc. but also to go from the device specifications to verifiable properties. This can be for example the voltage of the pacing for a pacemaker in a given configuration. This approach allows to verify real-world properties.

The healthcare domain has recently been plagued with cyber attacks in the form of ransomware attacks, where the intrusion often comes as a result of poor practices related to software patching and software inventory management. One mitigation that might help IT staff in deciding which software security updates that needs to be applied for securing medical devices is the introduction of a *Software Bill of Materials* (SBOM), where the manufacturer declares all software components in a device. In 2018 the FDA published a Medical Device Safety Action Plan where one of the proposed actions was to require medical device manufacturers to include an SBOM as part of their premarket submissions.

Securing devices to which an attacker might have physical access is hard. As mentioned in the Microsoft’s Ten immutable Laws of Security, “*If a bad guy has unrestricted physical access to your computer, it’s not your computer anymore.*” This is even more true for embedded medical devices, which usually do not come with as strong security defense mechanisms as computers. Indeed, adding strong hardware security to medical devices such as the Home Monitoring Unit also has a cost, and manufacturers might have to make a choice between security and costs, given that the money that is invested in the security of an HMU is not used for developing treatment functionality, which saves lives. In addition to this, there is also the race to market and the strict certification process that does not allow to easily make changes to an already approved design.

The industry is unfortunately not yet at the point where we can expect very strong cyber security in medical devices. As demonstrated by our research, basic security practices remains to be applied. A first step towards a more secure pacemaker ecosystem is the implementation of well-known best practices for hardware security, which, if they will not protect against all attackers, can surely raise the cost of an attack, and simply discourage many attackers. Guides such the Secure Design Best Practice Guide by the IoT Security Foundation [7] gives a checklist of se-

curity measures to be adapted to a product, already during the design phase. When it comes to securing the firmware, the OWASP foundations offers a project for Embedded Application Security [11] that should be taken into account.

6 Conclusion

In this paper, we presented the results of our research regarding the security of the pacemaker ecosystem of the medical device manufacturer BIOTRONIK, and the evolution of the security between the different versions of their Home Monitoring Units. We analyzed different versions of the devices and discovered vulnerabilities in all of them. While we confirm that the medical device manufacturing industry is still immature when it comes to the implementation of hardware security, we also point out that there is an evolution towards a safer ecosystem. We noticed improvements in the protocol used to communicate with the back-end infrastructure between the older and newer generations of devices. Our findings were reported to the vendor in a coordinated vulnerability disclosure process, resulting in an official security advisory, and will hopefully be used as input to improving the security of future devices.

ACKNOWLEDGEMENTS

We would like to thank *Anniken Wium Lie* with whom we collaborated on the network aspects of the project. We very much appreciate the contributions of *Éireann Leverett* that did some of the initial hardware testing to discover the HMU debug interfaces. Finally, we are grateful to *Snorre Aunet* and *Ingulf Helland* from NTNU who took time to help us solder a connector on the HMU.

This work was partially funded by *Reinforcing the Health Data Infrastructure in Mobility and Assurance through Data Democratization*, a five-year project (grant number 28885) under the Norwegian IKTPLUSS-IKT and Digital Innovation programme. The authors gratefully acknowledge the financial support from the Research Council of Norway.

References

- [1] Block, C. C. (2016). Muddy waters report - st. jude medical, inc. Technical report, Muddy Waters Capital LLC.
- [2] Bour, G. N. (2019). Security analysis of the pacemaker home monitoring unit: A blackbox approach. Master's thesis, NTNU.
- [3] Camara, C., Peris-Lopez, P., and Tapiador, J. E. (2015). Security and privacy issues in implantable medical devices: A comprehensive survey. *Journal of biomedical informatics*, 55:272–289.
- [4] CISA (2020). ICS Medical Advisory (ICSMA-20-170-05). <https://us-cert.cisa.gov/ics/advisories/icsma-20-170-05>. [Online; accessed 30-Sep-2021].
- [5] Denning, T., Fu, K., and Kohno, T. (2008). Absence makes the heart grow fonder: New directions for implantable medical device security. In *HotSec*.
- [6] Halperin, D., Heydt-Benjamin, T. S., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. H. (2008). Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 129–142. IEEE.
- [7] IoT (2020). Secure design best practice guide. [Online].
- [8] Justis- og beredskapsdepartementet, Helse- og omsorgsdepartementet (2005). Forskrift om medisinsk utstyr. <https://lovdata.no/dokument/SF/forskrift/2005-12-15-1690/%2FT1%2Ftextsection1-5#/T1/textsection1-5>.
- [9] Li, C., Raghunathan, A., and Jha, N. K. (2013). Improving the trustworthiness of medical device software with formal verification methods. *IEEE Embedded Systems Letters*, 5(3):50–53.
- [10] Marin, E., Singelée, D., Garcia, F. D., Chothia, T., Willems, R., and Preneel, B. (2016). On the (in) security of the latest generation implantable cardiac defibrillators and how to secure them. In *Proceedings of the 32nd annual conference on computer security applications*, pages 226–236.
- [11] OWASP (2020). OWASP embedded application security. [Online].
- [12] Rasmussen, K. B., Castelluccia, C., Heydt-Benjamin, T. S., and Capkun, S. (2009). Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 410–419.
- [13] Richards, R. (2015). Healthcare data breaches cost \$6 billion a year (infographic)t. [Online; posted 16 November 2015].
- [14] Rios, B. and Butts, J. (2017). Security evaluation of the implantable cardiac device ecosystem architecture and implementation interdependencies.
- [15] Rostami, M., Juels, A., and Koushanfar, F. (2013). Heart-to-heart (h2h) authentication for implanted medical devices. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1099–1112.
- [16] Savci, H. S., Sula, A., Wang, Z., Dogan, N. S., and Arvas, E. (2005). Mics transceivers: regulatory standards and applications [medical implant communications service]. In *Proceedings. IEEE SoutheastCon, 2005.*, pages 179–182. IEEE.
- [17] Schechter, S. (2010). Security that is meant to be skin deep using ultraviolet micropigmentation to store emergency-access keys for implantable medical devices.
- [18] Zheng, G., Shankaran, R., Orgun, M. A., Qiao, L., and Saleem, K. (2016). Ideas and challenges for securing wireless implantable medical devices: A review. *IEEE Sensors Journal*, 17(3):562–576.
- [19] Zheng, G., Yang, W., Valli, C., Qiao, L., Shankaran, R., Orgun, M. A., and Mukhopadhyay, S. C. (2018). Finger-to-heart (f2h): Authentication for wireless implantable medical devices. *IEEE journal of biomedical and health informatics*, 23(4):1546–1557.
- [20] Zheng, G., Zhang, G., Yang, W., Valli, C., Shankaran, R., and Orgun, M. A. (2017). From wannacry to wannadie: Security trade-offs and design for implantable medical devices. In *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, pages 1–5. IEEE.