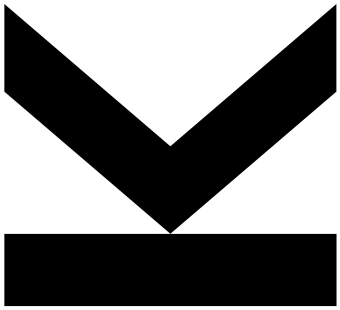
Ing. Michael Riegler, MSc
PhD candidate
LIT Secure and Correct
Systems Lab

P +43 732 2468 9540
michael.riegler@jku.at
jku.at

Final Research Report
Submitted to the Austrian Marshall Plan Foundation

RESILIENT MIDDLEWARE FOR SMART SECURITY IN MEDICAL DEVICES



By PhD candidate **Michael Riegler**
LIT Secure and Correct Systems Lab

Supervisor Home University:
Prof. Dr. Johannes Sametinger
Institute for Business Informatics – Software Engineering
Johannes Kepler University Linz, Austria

Supervisor Host University:
Prof. Dr. Jerzy W. Rozenblit
Department of Electrical & Computer Engineering
The University of Arizona, USA

JOHANNES KEPLER UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at
DVR 0093696

Acknowledgements

I am very grateful to the *Austrian Marshall Plan Foundation* for awarding me the *Marshall Plan Scholarship* for my research stay at the *University of Arizona*. I also appreciate the support from the *State of Upper Austria* for funding the *LIT Secure and Correct Systems Lab* and supporting me with the *Internationalization Program for Students*. Additionally, I thank the *National Science Foundation* under Grant Number 1622589 “Time-Centric Modeling of Correct Behaviors for Efficient Non-intrusive Runtime Detection of Unauthorized System Actions”, and the International Offices at the Johannes Kepler University Linz and the University of Arizona. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the supporting organizations.

My high appreciation goes to my supervisor at the *Johannes Kepler University Linz*, *Prof. Dr. Johannes Sametinger*, for encouraging me to go abroad and for his great support before, throughout, and beyond my stay in the United States.

Also, I am profoundly grateful to my supervisor at the *University of Arizona*, *Prof. Dr. Jerzy W. Rozenblit.* I immensely enjoyed our discussions at the Model Based Design Laboratory and his guidance during my time in Arizona and beyond.

I am thankful to everyone at the *Department of Electrical & Computer Engineering* for their hospitality and support during my time there, particularly *Nasser Albalawi* for his exceptional support and *Minisk Hong*, *Sofia Carreón*, *Vijay Nafria*, *Olu Sodipe*, and *Jonathan Biren* for their discussions, support, and input to my research.

I also sincerely appreciate my friends and family for supporting me during my stay in the US. I am very grateful for this exceptional experience.

Michael Riegler

Abstract

Network-connected medical devices and the *Internet of Medical Things* (IoMT) promise to improve patient care and efficiency. At the same time, they increase the security risk. The authors propose a resilient middleware to provide life-critical functionality even under adverse conditions. Medical devices require secure and self-protective functionalities that reduce attack surfaces and attackers' range of activity during malfunctions, attacks, or vulnerability exposures. Device recalls can be avoided by modeling and switching security modes at runtime, e.g., a degraded mode of operation with a smaller attack surface. When patches are provided and installed or attacks are over, the medical device can return to normal mode.

This paper presents ongoing work to make medical devices more secure by discussing current security and privacy challenges, how self-protective systems can overcome them, and the role of security modes in that context. We present a mode domain-specific language and a multi-modal architecture and show simulations on increasing medical device security and patient safety.

Table of Contents

1. Introduction

This final research report is a summary of our work during my research stay at the University of Arizona. In the following subsections, we present the problem definition, motivation, research agenda, goals, methodological considerations, and an overview of our publications. Section 2. gives an overview and theoretical background. In Section 3. , we describe common methods to secure medical devices and present our considerations about using a resilient middleware in Section 4. We extended this middleware to a self-protective healthcare system and show preliminary results in Section 5. Finally, we discuss our findings in Section 6. and draw our conclusions in Section 7.

1.1. Problem and Motivation

Through recent innovations in electronics and communication, medical devices ensure and increase in availability and efficiency. Active implantable medical devices like cardiac defibrillators, cochlear implants, brain stimulators, gastric stimulators, or insulin pumps support the well-being of patients. These devices are powered by batteries that have a lifespan of several years. Regular checks in the hospital are necessary to monitor devices and batteries. Remote monitoring enables patients to live more comfortably. Measured values of medical devices can get transferred automatically to hospitals via gateways. Medical doctors can analyze these measures and adapt configurations remotely. In case of patients' heart attacks, gateways can call emergency medical services and save lives.

Interfaces of medical devices and remote monitoring have advantages but also drawbacks. In 2019, 2 billion *Internet of Things* (IoT) devices, including many medical devices, were affected by 11 zero-day vulnerabilities, which, among other things, allowed remote code execution (Forbes, 2019). According to the *US Food and Drug Administration* (FDA, 2017), 465,000 US Americans were affected by cybersecurity vulnerabilities in implantable cardiac pacemakers. Unauthorized users could modify programming commands to rapidly empty batteries or harm patients with inappropriate pacing. Fortunately, no surgery was necessary to mitigate the situation. Firmware updates were sufficient. These updates were recommended but not mandatory. The time between a vulnerability's disclosure and the availability of updates is crucial. Attackers can write exploits, i.e., software to take advantage of vulnerabilities. To mitigate such threats, a research team around *Prof. Dr. Jerzy W. Rozenblit* and *Prof. Dr. Roman Lysecky* from the University of Arizona and *Prof. Dr. Johannes Sametinger,* my supervisor from the Johannes Kepler University Linz, have proposed to build systems with several operational modes (Rao et al., 2019). Each mode consists of several tasks. If malware is detected or known, modes can be switched to have a smaller attack surface with a limited range of activity for attackers, and perhaps limited functionality of the device.

During my research stay at the University of Arizona, I could build up on this holistic approach and showed how this work can practically mitigate cyber threats. The *Marshall Plan Scholarship* was a great opportunity for me to connect with *Prof. Rozenblit* and his teams at the *Department of Electrical & Computer Engineering* (ECE) and the *Model Based Design Laboratory* (MBDL) and get more profound knowledge in the domain of medical device security. We plan to make the next generation of connected medical devices more secure and contribute to the technical development in this domain.

1.2. Research Agenda

It is only a matter of time until connected medical devices will get attacked. Our research team, led by *Prof. Rozenblit* from the University of Arizona and my supervisor, *Prof. Sametinger* from the Johannes Kepler University Linz, has suggested a solution to address the risks mentioned in Section 1.1. The approach involves developing systems with multiple operational modes, as proposed in a study by Rao et al. (2019). Each mode is comprised of several tasks. Switching these modes can protect endangered systems if malware is detected or known to exist. These systems should automatically or manually switch to a mode with a limited attack surface and a limited range of activity for attackers. Attack surface reduction may come at the expense of limited functionality. However, mode switching can overcome the time between a disclosed zero-day vulnerability and the availability of a security patch. Therefore, healthcare organizations and even patients get the capability to actively protect themselves by switching to a secure and safe mode.

The concept of mode switching is well known in domains like, for example, aviation. Airplanes have a parking mode, a taxiing mode, a take-off mode, a manual and automatic flying mode, a landing mode, and an emergency mode. Such multi-mode systems manage complexity and divide systems into modes of operation in the automotive domain. For example, self-driving cars may use manual, adaptive cruise control, parking, or emergency braking modes (T. Chen & Phan, 2018, 2018). Each mode consists of a set of functionalities and a system configuration as well as different control goals. Considering security as condition to switching modes is quite new in the scientific discussion and not yet shown practically.

The last missing link is a resilient middleware between the real-time operating system, the risk detection system, and the healthcare application. No matter how well a system has been designed, vulnerabilities that are still unknown or not resolved can always become a risk. Therefore resilience is essential. Resilient systems protect their critical assets by using methods to detect adversities, mitigate them and recover from them. To develop such a middleware, we must take several steps and answer the following research questions:

- What are common mode-switching protocols, and how can they be applied to mitigate risks from a security perspective?
- How can mode switching work within medical devices? What are the challenges?
- How does mode switching make a medical device more secure and resilient?
- Which types of threats can be mitigated by mode switching?
- Which recommendations can be deducted for medical device manufacturers (MDM) and healthcare delivery providers (HDO)?

1.3. General Goals

- Improve security in computer-based technologies for medical devices and make them more resilient against cyberattacks
- Provide a mechanism for software safety and security risk mitigation in medical devices
- Identify and develop methods of smart mode switching with an adaptive risk assessment which works on medical devices
- (Medical) device prototype/simulation to show how smart mode switching can work

1.4. Methodological Considerations

A middleware between the operating system, the real-time risk detection system, and the healthcare application is the last missing link. Therefore, **action research** was conducted to analyze existing mode-switching protocols and solutions and discover how they can be applied to ensure security and resilience. **Modeling modes** with a domain-specific language simplified the definition of multi-modal systems. A **simulation** about medical devices shows how mode switching can overcome the time between a disclosed zero-day vulnerability and a security patch. An **evaluation** with several attack scenarios and simulated attacks shall show the effectiveness of this new approach.

1.5. List of Publications

During my research stay, my colleagues at the University of Arizona and Johannes Kepler University Linz and I were working on the following publications:

- Riegler, M., Rozenblit, J.W., Sametinger, J.,: "**Context-Aware Security Modes For Medical Devices**", *2022 Annual Modeling and Simulation Conference (ANNSIM)*, San Diego, CA, USA, 2022, pp. 372-382, https://doi.org/10.23919/ANNSIM55834.2022.9859283.

- Riegler, M.; Sametinger, J., Schönegger, C.: "**Mode Switching for Secure Edge Devices**" in: *Database and Expert Systems Applications - DEXA 2022 Workshops*. DEXA 2022. Communications in Computer and Information Science, vol 1633. Springer, Cham, https://doi.org/10.1007/978-3-031-14343-4_32.

- Albalawi, N.S., Riegler, M., Rozenblit, J.W.: "**Towards Strategies for Secure Data Transfer of IoT Devices with Limited Resources**" in: *Database and Expert Systems Applications - DEXA 2022 Workshops*. DEXA 2022. Communications in Computer and Information Science, vol 1633. Springer, Cham, https://doi.org/10.1007/978-3-031-14343-4_30.

- Riegler, M., Sametinger, J., Vierhauser, M., Wimmer, M.: "**A model-based mode-switching framework based on security vulnerability scores**", *Journal of Systems and Software*, 2023, 111633, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2023.111633.

- Riegler, M., Sametinger, J., Vierhauser, M.:" **A Distributed MAPE-K Framework for Self-Protective IoT Devices**", *accepted for publication at 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, Melbourne, Australia, 2023.

- Riegler, M., Rozenblit, J.W., Sametinger, J.,: "**Self-protective Healthcare Systems with Medical IoT Devices**", *submitted for publication*, 2023.

2. Theoretical Background

Medical devices that are actively implanted, such as insulin pumps, brain stimulators, cardiac defibrillators, cochlear implants, and gastric stimulators, are crucial for maintaining patient health. These devices are usually battery-powered and have a lifespan of several years. It is essential to conduct regular checks in hospitals to monitor these devices and their batteries. Remote monitoring and interconnected devices allow better patient care, happier patients, and cost reductions (Volterrani & Sposato, 2019). Therefore more and more medical devices get connected. Measured values of so-called *Internet of Medical Things* (IoMT), like heart rate, pulse rate, blood pressure, body temperature, and others, can be transferred wirelessly and automatically to healthcare service providers (Suresh et al., 2020). If needed, authorized medical personnel can analyze the data and adapt configurations remotely. Additionally, IoMT devices can automatically alert medical personnel if certain values fall below or exceed a predefined threshold, which is more efficient than traditional personal emergency response systems with manual alerts. The emergence of COVID-19 has increased the development and adoption of IoMT, which combines medical devices, sensors, applications, and services to enable remote medical care delivery and reduce in-person visits, especially during times of lockdown. Because of the growing IoMT market and the introduction of the 5G mobile communication standard digital interconnectivity is unstoppable. According to (Reports And Data, 2021), the global market value of IoMT will reach over 260 billion US dollars in 2027.

Nevertheless, these connections go hand in hand with security vulnerabilities and potential threats to these devices. According to Gartner (2021), by 2025 "cyber attackers will have weaponized operational technology environments to successfully harm or kill humans". The interfaces of the medical device and remote monitoring can have drawbacks. In 2019, 11 zero-day vulnerabilities affected 2 billion IoT devices, including many medical devices, which among other things, allowed remote code execution (Forbes, 2019). These vulnerabilities permitted remote code execution and resulted from flaws in software development kits used for Wi-Fi, Bluetooth, and *Bluetooth Low Energy* (BLE) connections. These issues could allow attackers to cause deadlocks, crashes, buffer overflows, or completely bypass security (Garbelini et al., 2020; Yaqoob et al., 2019). Hundreds of thousands of US Americans were affected by cybersecurity vulnerabilities in implantable cardiac pacemakers in 2017 (FDA, 2017). These vulnerabilities could allow unauthorized users to modify programming commands, resulting in rapid depletion of the battery or harm to the patient through inappropriate pacing. Fortunately, a firmware update was sufficient to address the issues, and no surgery was required. Although recommended, the update was not mandatory. More than a hundred devices were, and some still are affected by life-critical cyber-attacks (Alsubaei et al., 2017; Halperin et al., 2008; Kaplan, 2011; Li et al., 2011; Paul et al., 2011; Sun et al., 2018; Yaqoob et al., 2019). Over the past years, the FDA has recalled multiple medical devices due to possible cybersecurity threats. According to Claroty (2022), there has been a 110% increase in vulnerability disclosures since 2018. Recent reports have also shown that over 75% of more than 200,000 infusion pumps have security gaps, which could result in serious issues such as privacy breaches, battery depletion, malfunctioning, extortion, remote assassination with manipulated settings, and even fatal doses of medication (Palo Alto Networks, 2022). Some medical device manufacturers use off-the-shelf hardware and software, making them vulnerable to exploits such as *URGENT/11, SweynTooth, Ripple20, AMNESIA:33, BadAlloc, Nucleus:13, Log4Shell*, and others.

2.1. The Window of Vulnerability

The time between a vulnerability becoming known to the public and the availability of an update or patch is very crucial. Stopping a pacemaker or another life-critically medical device in case of a cyber-security threat is no option. Medical devices are certified through quality assurance processes. Due to safety requirements developing and distributing updates can take a comparably long time. Healthcare organizations may have to wait helplessly until the manufacturer releases the update, otherwise the guarantee is denied. Only then they can invite patients and install the update. However, as we know from our operating systems, there is still a risk of complete loss of the device functionality after the update. In the meantime, hackers can write an exploit to take an advantage of the vulnerability. Moreover, what if the manufacturer has discontinued support or if the manufacturer no longer exists? In 2022, the *US Federal Bureau of Investigation* (FBI, 2022) warned healthcare organizations about the increasing vulnerabilities of outdated and unsecured medical devices. Default configurations and lack of security features and updates pose a significant risk to hospitals and patient safety.

2.2. Limited Resources

Due to space and cost limitations, IoMT devices are expected to have restricted resources compared to traditional IT systems (Pisani et al., 2020). Some of the devices have only battery power, limited memory, lower processing speed, and low network bandwidth. Therefore, IoMT devices are connected to either edge or cloud computing, which provides adequate resources such as computation, memory, and storage. This approach, known as computation offloading, enables IoT devices to consume less power since most of the computation and storage are performed in the edge or cloud (Samie et al., 2016). Lightweight communication protocols like *MQTT* and the *Constrained Application Protocol* (CoAP) are used in low-power and lossy networks (Espinoza et al., 2017). However, the limited resources have a significant impact on security. IoMT may not have the ability to implement robust security measures. This makes them vulnerable to cyber-attacks, data breaches, and other security threats.

2.3. Safety and Emergency

Medical devices, especially implanted ones with batteries, have limited resources and limited functionality and thereby offer special security challenges (Sametinger et al., 2015). Safety is always the most important goal in protecting the environment, patients, and medical staff from the device. Therefore manufacturers must ensure that devices do not harm patients. The opposite of safety is security which aim is to protect a device from the environment. However, safety is at stake when security is weak. For example, if an attacker gets access to a pacemaker and changes the clock rate or triggers an emergency shock that harms the patient. In that case, a security issue becomes a safety issue.

Imagine the situation if you are on vacation far from home and need medical help. In an emergency, security mechanisms must not endanger patients' lives (MDCG, 2019). Availability and a fail-safe state are more important for these medical devices than confidentiality and integrity. However, open access can be a risk and contradicts the idea of security by default. Under normal conditions, unauthorized persons should not be able to make any changes. Therefore, we have to align our system architecture to fulfill these requirements. Medical devices should consider the context, automatically detect critical situations and adapt the behavior.

3. Securing Medical Devices

In this section, we show the security and privacy challenges of medical devices and present guidance documents to overcome these challenges and fulfill regulations. Additionally, we present how mode switching and security modes can improve the defense-in-depth strategy.

3.1. Security and Privacy Challenges

Various authors (Ajagbe et al., 2022; Elhoseny et al., 2021; IMDRF, 2020; Kagita et al., 2022; Sametinger et al., 2015; Sun et al., 2018; Thomasian & Adashi, 2021; Yaacoub et al., 2020) have discussed the challenges in the medical domain, including confidentiality, integrity, availability, reliability, safety, privacy, secure communication, software and hardware aspects, intrusion detection and reaction, formal methods, resource constraints, non-technical aspects, and organizational and regulatory issues. According to the chronology of medical device security (Burns et al., 2016), considering threats and zero-day vulnerabilities in medical devices are necessary throughout their entire useful life, from design, development, and distribution through maintenance and maintenance decommissioning.

It is important to speed up the development and distribution of updates to fix vulnerabilities. However, updating medical devices can be challenging due to their long lifespan, the need for legacy systems, and the lack of automatic update mechanisms. Medical device manufacturers and healthcare delivery organizations should deliver updates from trusted sources and ensure they do not introduce new vulnerabilities. Formal methods must be used to verify the correctness of software updates, and resource constraints must be considered. Regulations may also slow down the process. Failing to fix vulnerabilities leaves medical devices susceptible to attacks, as attackers can exploit the vulnerabilities. (Sametinger & Rozenblit, 2016)

Confidentiality must be maintained through identity and access management, and sensitive medical data and device programming must be protected, particularly remote access. Audit controls can help monitor and detect suspicious access and changes. It is also important to avoid storing hard-coded passwords, keys, and other sensitive information unencrypted on the device, as compromised or stolen devices can threaten other devices. Data integrity must be protected through hashing algorithms and checksums because false or modified data may result in an incorrect diagnosis, treatment, or medication and risk the patient's health. Medical devices must be operational and reliable to deliver clinical functions to patients. Safety is a top priority, as IoMT should not harm patients in any way. IoMT devices should be able to detect, resist, respond, and recover from attacks. Availability must be maintained to prevent *Denial of Service* (DoS) attacks from overloading a device, causing battery depletion, or disrupting critical medical functions. Privacy must be protected to prevent violating sensitive individual health and behavior data. Violations may seriously affect patient safety and the reputation of HDOs and MDMs, potentially resulting in penalty payments. Data encryption and key management are necessary to provide secure and reliable communication and prevent *Man-in-the-Middle* (MITM) attacks like packet modification or replay. Software and hardware security must be ensured, and intrusion detection and reaction mechanisms must be in place to prevent malicious activities. Finally, non-technical, organizational, and regulatory aspects must be considered, and backup and recovery plans must be in place to mitigate ransomware attacks.

3.2. Guidance Documents

In the United States, the *Food and Drug Administration* (FDA) provides pre- and post-market guidance on building secure medical devices and using a *Bill of Materials* (BOM) to monitor supply chain problems and vulnerabilities of commercial, open-source, and off-the-shelf hardware and software (FDA, 2016, 2022). BOMs simplify monitoring and reduce response time in case of incidents or vulnerabilities. The *NIST Cybersecurity Framework* recommends considering functionality to prevent unauthorized use, loss of confidentiality, integrity, availability, and, thus, patient harm (Barrett, 2018). Using the five core functions: *identify, protect, detect, respond, and recover*, they present guidelines to manage and reduce cybersecurity risks. MITRE provides a playbook to identify and mitigate medical device threats (Bochniewicz et al., 2021). Safeguards and other protection methods are needed to limit access and ensure that only trusted and authorized users and devices execute safety-critical commands.

In the European Union, the *Medical Device Coordination Group* (MDCG, 2019) offers guidance for developing and manufacturing medical devices in compliance with state-of-the-art and meeting the *Medical Device Regulation* (MDR). They suggest a Defense-in-Depth strategy during the product lifecycle. Medical device manufacturers must consider risk management, information security, and IT security measures like protection against unauthorized access and foreseeable misuse across the life cycle. The *International Medical Device Regulators Forum* (IMDRF, 2020) provide design principles and best practices, such as secure communication and software maintenance, to assist all stakeholders in their collective duty to ensure medical device security. They mention pre- and post-market considerations for secure architectures, testing, vulnerability remediation, and incident response. Additionally, the *Association for the Advancement of Medical Instrumentation* (AAMI), the *Healthcare Sector Coordination Council* (HSCC), the *International Society of Automation* (ISA), and the *International Electrotechnical Commission* (IEC) established industry standards and guidance like AAMI TIR57, AAMI TIR97, HSCC Model Contract Language, and ISA/IEC 62443.

3.3. Mode Switching

Our literature review (Riegler & Sametinger, 2020) discusses security-related findings on modes and mode switching. Modes are used to ensure proper operation in case of failures. They are a logical framework combining system states, providing specific functionalities, and facing different security risks. Multi-mode systems divide and manage complexity, having specific configurations and behaviors. For instance, airplanes have various modes for different functionalities like taxiing, starting, flying, and landing. Switching between these modes has implications for functionality and security. Nuclear power plants utilize various modes to simplify configurations and decrease complexity, such as for maintenance and ensuring a secure shutdown during emergency situations (US Nuclear Regulatory Commission, 1995). Modes have a crucial function in the engineering of system resilience and security, as noted by Firesmith (2019), the Joint Task Force Transformation Initiative (2014), and Ross et al. (2016). When confronted with errors, failures, or attacks, systems can smoothly transition to a degraded or safe mode while still offering a minimum level of service instead of shutting down the system. From that degraded mode, systems may recover from the disruptions to a fully functional mode.

The German Federal Office for Information Security (BSI, 2018) has established different modes for network-connected medical devices, including modes for medical operation, configuration, and technical maintenance, in their cyber security requirements. For secure and trustworthy systems, Ross et al. (2016) propose various methods to address potential disruptions, hazards, and other threats. They outline different modes of operation, including initialization, normal/operational/runtime, alternative, degraded, secure, standby, maintenance, training, simulation, test, recovery, shutdown/halted, and others. Each mode is characterized by its unique behavior, security settings, and prescribed transitions to other modes. Rao et al. (2019) propose a trustworthy multi-mode framework for life-critical systems. Modes are also used for secure data transmission (Almazyad et al., 2020) and power management (Alemzadeh et al., 2013; Samie et al., 2019). Easttom & Mei (2019) propose a software shim with a normal and an emergency mode to protect implanted medical devices, switching to the emergency mode when detecting an anomaly and allowing only data synchronization.

3.4. Security Modes

Medical devices can either be secure or insecure, depending on the existence and severity of vulnerabilities, regardless of their sensitivity, impact, or exposure. Security scores were introduced to classify devices from a privacy and safety perspective (Sametinger & Steinwender, 2017). A defense-in-depth strategy with security modes is suggested for these devices, enabling a reduction of exposure when vulnerabilities are detected or exploited. Switching modes and, thus, the device's security scores by the manufacturer, healthcare delivery organization or even patients can reduce the attack surface. If devices have the sufficient processing power and power supply, they can also react to their environment and switch to a more secure mode upon detecting potential intrusion.

Context awareness can take various forms (G. Chen & Kotz, 2000). Active context-awareness involves adapting the behavior of a medical device based on the discovered context, such as the device behaving differently in case of an emergency. Passive context-awareness involves informing device users about alerts for a specific brand or model. Medical devices can directly contact a server to obtain security context, but resource limitations may make this difficult. Home monitoring systems can provide this service and propagate the information to an implanted device. Patients can confirm access to a medical device with a button on the home station. The device or home station can notify the healthcare organization and/or device manufacturer about mode switches. Anomaly detection is the most effective protection through active context-awareness, where a device recognizes abnormal behavior and initiates countermeasures, like switching to a more secure mode (Carreon et al., 2021; Lu et al., 2015; Lu & Lysecky, 2019).

The window of vulnerability opens when a vulnerability becomes known and can only be closed with a patch. Medical device manufacturers control this process and shortening the window of vulnerability is the goal. Patients and healthcare organizations have little influence. Medical devices that can switch to a more secure mode can provide resilience during the window of vulnerability and then switch back to normal. This self-healing process must be implemented carefully to prevent abuse. For example, *Trusted Platform Modules* (TPMs) increase healing time after each failed attempt (Trusted Computing Group (TCG), 2022).

4. Resilient Middleware

To overcome limitations in accessing medical device hardware and software, we plan to model and simulate specific medical devices. We focus on a middleware between the different applications, services, and systems. The middleware should mitigate cyber-attacks, minimize downtime, and keep systems operational. By implementing multiple modes, we want to provide a failover mechanism in case of attacks or vulnerabilities. Continuously monitoring the components and the environment is needed to detect issues and proactively address them before they become critical. We will model a subset of modes and run simulations with real-world vulnerabilities and attacks to test *self-protection* and *self-healing* processes. We successfully applied this approach to web applications, as demonstrated in a two-year simulation (Riegler, Sametinger, Vierhauser, et al., 2023). We retrieved *Common Vulnerability Exposures* (CVEs), severity scores based on the *Common Vulnerability Scoring System* (CVSS), provided patches based on the systems' BOM, and used this information to make mode switch decisions.

4.1. Mode Modeling

In (Riegler, Sametinger, Vierhauser, et al., 2023), we created a *Mode Domain-Specific Language* (MDSL) to establish and simplify the process for a multi-modal architecture for resilient system protection. Activating and deactivating components manually or creating multiple scripts for mode switching can be a cumbersome and error-prone process, particularly when dealing with multiple versions or configurations of components. To alleviate this, we provide a declarative specification of components, modules, and actions, which simplifies the task of defining and maintaining these elements. When combined with model-to-code transformation, we can automatically generate executable code or scripts based on the specified modes and modules, making it easy to adopt new components and define additional modes or updated versions of existing modules without manually updating any code or scripts. We defined different modes by varying software components and versions from multiple vendors. Switching to another mode and, thus, another software component or version can mitigate known vulnerabilities. Figure 1 provides an excerpt of our MDSL. Each mode has a description, a priority, specific start/stop actions, and uses specific software components. Additionally, each mode utilizes software versions, which can be affected by vulnerabilities and lead to a mode switch. If two or more modes present equal risks, we determine the priority. For instance, Figure 2 shows the mode definition for *ApacheWithPhp*. This mode refers to the *Apache Webserver* and extends the super mode *Apache* with the open-source server-side scripting language PHP. The mode has the priority one, and specific start and stop actions for Apache modules and Linux services with parameter values. Additionally, the used software version 7.3.5 of PHP is specified. In our future development, software versions may be detected automatically.

```
'Mode' name=ID ('extends' superType=[Mode])?
'description' description=STRING
'priority' priority=INT
'startActions' (startActions+=Action  (',' startActions+=Action)*)?
'stopActions' (stopActions+=Action  (',' stopActions+=Action)*)?
'usesSoftware' (usesSoftware+=[Software] (',' usesSoftware+=[Software])*)?
```

Figure 1: Abstract Mode Definition

```
Mode ApacheWithPhp extends Apache
description "Static_HTML_and_dynamic_PHP_pages"
priority 1
startActions enableApacheMod("proxy_fcgi","setenvif"), enableApacheConf("php7.3-fpm"),
     startLinuxService("php7.3-fpm")
stopActions stopLinuxService("php7.3-fpm"), disableApacheConf("php7.3-fpm"),
    disableApacheMod("proxy_fcgi","setenvif")
usesSoftware php_7_3_5
```

Figure 2: Mode Definition ApacheWithPhp

We used two years of historical vulnerabilities and patches to analyze how mode switching can reduce the window of vulnerability. Each mode, software, and version had different vulnerabilities over time. In a web server case scenario, we reduced the window of vulnerability from 536 to 8 days with 7 to 11 mode switches, resulting in zero known risk in 98.9% of the analyzed time. Figure 3 shows a comparison of summed-up vulnerability scores over two years for *ApacheWithPhp*, *NginxWithPhp*, and *Mode-Switching*.

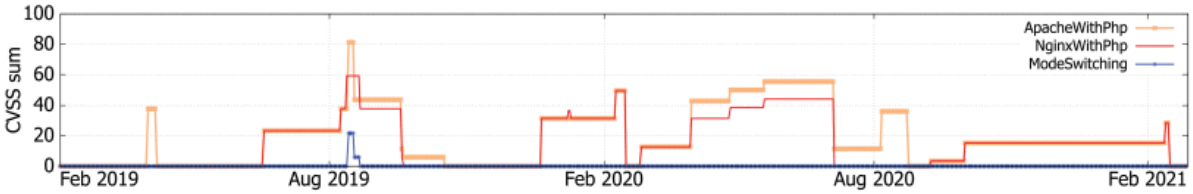


Figure 3: Vulnerability Scores of Web Server Scenario

Our plan is to utilize our mode domain-specific language for a cardiac pacemaker scenario and incorporate rules and events to enhance the mode-switching settings. We aim to monitor and manage various multi-mode scenarios and explore ways to improve the security and resilience of medical devices. As pacemakers lack sufficient computing power for threat detection, we envision triggering mode switches externally. Regular doctor check-ups, conducted every six months, present an opportunity to check for abnormalities, battery status, and optimize programming. These check-ups could also initiate security mode switches when necessary. Alternatively, home stations can prompt mode switches in a timelier fashion.

4.2. Multi-Modal Architecture

Figure 4 presents an overview of our proposed multi-modal architecture for medical devices in (Riegler et al., 2022). The architecture consists of three main components: the *configuration part* utilizing the mode domain-specific language (orange), the *mode control part* (blue) within the medical device, and the *inventory part* (yellow). The *Operator's* first step (1) is the definition of the desired modes with a *System Mode Description* using our MDSL and saving it in the inventory. Based on that, the *System Mode Configuration* is automatically generated (2). The *Mode Control* component uses and runs the *System Mode Configuration*, analyzes events, and can execute mode switches, which may be triggered by a Log File Analyzer (11), an Intrusion Detection System (11), or a Vulnerability Analyzer (5). The *Vulnerability Manager* (red) automatically gathers information (3), such as CVEs, patches, and exploits from public databases and vendors for the specified parts of the system in the *System Mode Description*. New CVEs, patches, exploits, and changed CVSS scores are forwarded (4) to the *Event Analyzer*. We use the CVSS scores of all software components and modes to prioritize them. The scores for each mode are then used to determine whether a mode switch is necessary.

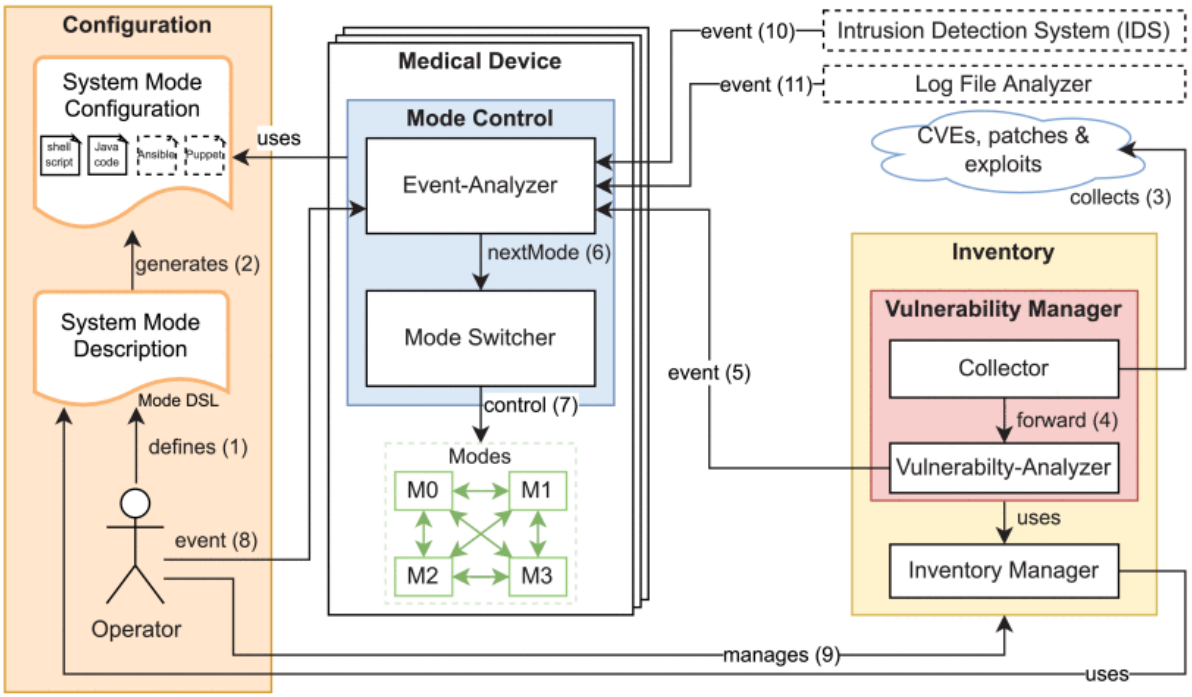


Figure 4: Conceptual Overview of our Multi-modal Architecture for Medical Devices

Events may be automatically sent to the medical device based on rules and policies or require human-in-the-loop for further investigation. Thresholds can be employed to prevent frequent mode changes initiated by intrusion detection systems or log file analyzers. The *Event Analyzer*, integrated into the medical device, assesses events and determines if the current mode is appropriate. If a mode switch is necessary, the *Mode Switcher* is called (6) to execute the required actions to initiate the new mode (7). When certain services are used in multiple modes, we keep them running during mode switches to minimize potential downtime and increase efficiency. This partial stop/start approach is designed to reduce interruptions. Additionally, administrators have the ability to adjust mode priorities and manually enable or disable modes for precautionary reasons, such as when vulnerabilities or exploits are discovered but not yet documented, or when unexpected events occur. For safety and to reduce the attack surface, manual mode switches can be required for specific medical devices, or a set of medical devices triggered by the operator (8). The operator manages (9) the inventory of various medical devices and the *Vulnerability Manager's* settings.

4.3. Mode Simulation

Medical devices have few publicly known CVE entries, and manufacturers may not disclose vulnerabilities. According to Medcrypt (2022), medical advisories by the *Industrial Control Systems Computer Emergency Response Team* (ICS-CERT) of the *US Cybersecurity and Infrastructure Security Agency* (CISA) increased by 490% after the FDA's *Postmarket Cybersecurity Guidance* in 2016. User-authentication mismanagement and code defects are common root causes. Researchers discovered 71% of vulnerabilities in 15 device types. Infusion pumps, imaging software, patient monitors, and cardiac rhythm management were the most affected, but manufacturers may only provide detailed information to registered customers or keep incidents confidential.

We examined *ICS-CERT medical advisories* (ICSMA) and the vulnerabilities of medical devices and selected ICSMA-21-187-01 as an example (CISA, 2022). The medical advisory contained a total of 11 vulnerabilities for the *Philips Vue Picture Archiving and Communication System* (PACS). The root cause of most vulnerabilities was code defects and insecure third-party libraries. It took more than two years on average to provide a fix for these vulnerabilities, including a publicly known one that took over nine years to fix. Four third-party vulnerabilities with publicly known exploits were also patched up to four years later. The timeline and vulnerability scores are shown in Figure 5.

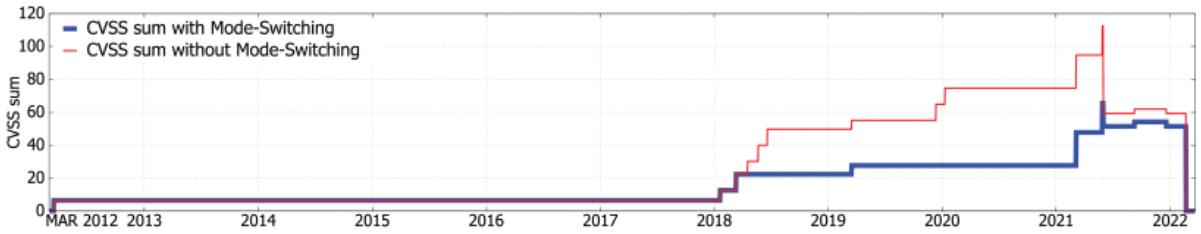


Figure 5: Vulnerability Scores of Philips Vue PACS ICSMA-21-187-01

We propose a multi-modal system to detect and mitigate third-party vulnerabilities until a patch is available. Instead of relying on a single software program like *7-Zip* or *Redis*, we can use alternative implementations like *gzip* or a different key-value database. We can also use different protocols for data transfer, such as HTTPS instead of the *Apache JServ Protocol*. This approach adds complexity but provides operational flexibility. In our example, switching between third-party software reduces the CVSS sum on 1404 of 3760 days, resulting in a 41% average decrease from 28.2 to 16.6. We can also switch modes between different configurations of a single implementation, not just between implementations.

4.4. Prototypical Implementation

To provide a proof-of-concept for our mode switching approach and to experiment, we developed a Java prototype implementation of our multi-modal framework in Riegler et al. (2023). This implementation includes the domain-specific language for defining modes, services, actions, the mode control components, and the code generator mentioned in Figure 4. We focused on the model-based generation of the system configuration and mode control.

In order to implement our mode domain-specific language, we used the *Eclipse Modeling Framework* (EMF) as a meta-model and provided a textual syntax using the *Xtext framework*. To ensure valid system configurations, we implemented a validation component that verifies package names according to the operating system detected or specified, preventing duplicates and logical errors such as self-inheritance. We also provided several predefined default actions to simplify defining a system description, such as starting and stopping specific services without manually defining these commands for each operating system. Once a system description has been defined or modified using our mode domain-specific language, the respective system configuration can be automatically created within the *Eclipse Editor* or by running the *Mode Switching Framework* on the command line. We used *Xtend* to generate Java code from the file with the extension ".mdsl" to define modes and actions. We then create the Java bytecode from it. For the evaluation described in Riegler et al. (2023), we developed a crawler to collect corresponding CVEs, and patches, which can trigger mode switches once a vulnerability is reported.

5. Self-Protective Healthcare Systems

In Riegler, Sametinger & Rozenblit (2023), we extended the perspective of our multi-modal architecture for interconnected medical devices, including medical sensors, applications, and services that enable remote medical care delivery and reduce the need for in-person visits. With the increasing number of medical and IoMT devices, safeguarding and ensuring their security requires a more comprehensive approach beyond securing a single device. Given that these medical devices are interconnected, network-based security measures can be utilized to enhance their security. The detection of certain anomalies and attacks may only be possible or easier when multiple IoMT devices are employed alongside a central control component.

According to de Lemos et al. (2013), self-adaptive systems aim to automatically configure, reconfigure, optimize, and update themselves at runtime. These systems are typically managed and controlled using the *MAPE-K feedback loop* (Arcaini et al., 2015; Kephart & Chess, 2003), which is widely used in different domains such as autonomous vehicles, IoT applications, and service-based systems. Monitoring, analysis, planning, and subsequent adaptation are fundamental in these systems to adapt the architecture, optimize performance and response time, ensure reliability, and even address security concerns in some cases. In that context, we focus on the self-protection of healthcare systems and IoMT devices. Several approaches in this field have addressed security concerns, such as the approach presented by Tomić et al. (2018), which detects attacks on network communication by operating on the network level and enabling data routing adaptations. Abie et al. (2010) describe a messaging infrastructure that adapts and targets security vulnerabilities to improve reliability and robustness, with a focus on message-oriented middleware. Van Landuyt et al. (2021) propose reflective threat modeling to support adaptive security at runtime. Although these approaches enable runtime adaptation and consider security concerns to some extent, they do not provide a flexible mode concept and a DSL as proposed in our multi-mode architecture.

5.1. Architecture

As depicted in Figure 6, our proposed architecture for *Self-protective Healthcare Systems* (SPHS) follows the principle of divide and conquer. The decision-making process is decentralized as much as possible and centralized as required. A *Manager* application on the severs-side continuously monitors the connected *IoMT devices* using its inventory. Thereby the system can provide enhanced visibility and situational awareness for the *Operator*. After analyzing the situation, the *Operator* can plan and execute adaptations to achieve the system goals and automate repetitive tasks and decisions. In addition, the *Manager* centrally monitors public databases, such as CVEs, medical advisories, safety communications, product alerts, warnings, and recalls, as presented in our previous work (Riegler, Sametinger, Vierhauser, et al., 2023). Based on this information, the *Manager* can automatically send adaptation messages to the *IoMT devices* to modify their behavior, such as blocking IP addresses or switching their mode (M0-M3). Likewise, the *Operator* and the *Patient* can also perform manual modifications. The architecture is designed to defend against attackers at multiple levels, with anomaly and attack detection and reaction implemented on different levels. The *IoMT devices* monitor and affect their environment, and decisions about further actions are made either automatically by the *Manager* or manually by an *Operator*.

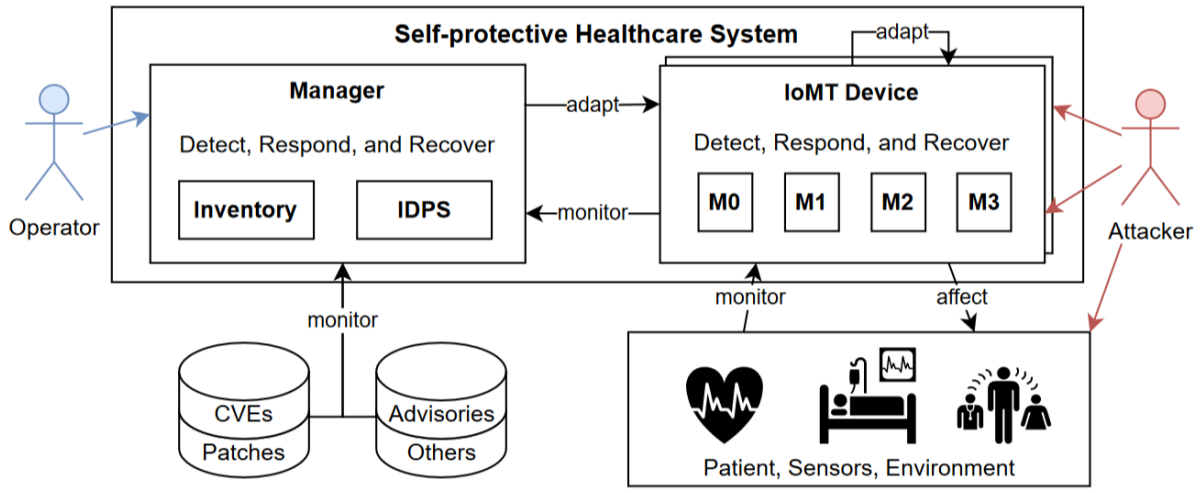


Figure 6: Self-protective Healthcare System Architecture

The architecture leverages our multi-modal approach in Riegler et al. (2022) and extends it with an *Intrusion Detection and Prevention System* (IDPS) on both server- and IoMT devices-side. Through local firewall rules, IoMT devices can block attackers' IP addresses after multiple failed attempts, like brute-force attacks on credentials or encryption keys. If these attacks persist and have an impact on the device's availability or battery life, the IoMT device can adapt itself by switching to a more restrictive mode. We recommend a low-power mode with limited functions to reduce the attack surface and extend battery life. An activity sensor or timer can trigger a switch to a mode with more functionality. Additionally, when the IoMT device connects to the *Manager*, switching to a high-security mode can provide an encrypted channel, making the device more resilient against Man-in-the-Middle attacks. To recover from attacks and reestablish functionalities self-healing capabilities can be used (Carreon-Rascon & Rozenblit, 2022).

5.2. Sample Scenario

We had a closer look at *Automated insulin delivery* (AID) systems. These systems can be used both in hospitals and at home for patients with chronic diseases such as diabetes (Medtronic, 2022a). These systems consist of wearable devices that monitor vital signs and continuously monitor blood glucose levels, wirelessly connected medicine pumps, and handheld devices or smartphones for local control and connection to the healthcare delivery organization. Based on device settings, patient history, and the current condition, the handheld analyzes the data and may adapt the settings. For example, if the blood glucose level changes, the system decides to increase or decrease the dose of medication. Within a specific threshold, this process is automated as a closed loop. In the *Internet of Medical Things* context, this scenario is extended with information transfer to the healthcare delivery organization for remote monitoring and reconfiguration by a physician, as described by Rao et al. (2022). If a measured value is outside the threshold, the operator at the healthcare delivery organization gets a notification, reviews the data and the generated recommendation, and may adapt the settings.

There are several commercial and non-commercial automated insulin delivery systems available. However, vulnerabilities in such systems can compromise patient safety. For example, Medtronic (2022b) alerted patients about a vulnerability in their *MiniMed 600 series* protection mechanism that could compromise the communication, allow unauthorized users to change the insulin delivery, and “potentially lead to seizure, coma, or death”. To address this issue, our *Self-protective Healthcare System Architecture* could be used. This architecture would simplify the steps for patients by allowing them to switch between connected and disconnected modes. In the disconnected mode, the pump works offline and considers only pre-defined presets and manual changes using the switches on the physical hardware. The disconnected mode is also the fallback mechanism if the connection to other devices is lost. In the connected mode, the pump would consider the information of connected sensors and automatically adapt the medication dose.

The *Self-protective Healthcare System Architecture* includes a manager that recognizes vulnerabilities, notifies the operator, and suggests actions to adapt the IoMT devices. The inventory allows the operator to notify patients directly at the device and obtain consent before executing interventions. Additionally, security-concerned patients can manually switch from the connected to the disconnected mode as needed.

Another recommendation would be to connect or link devices only in private places. Only predefined connections to trusted devices are allowed in the connected and protected mode, but no new ones to reduce the attack surface. The lightweight intrusion detection and prevention system on the IoMT device could analyze the traffic and data from connected devices, notify the patient and the operator about abnormal behavior, and automatically delete the suspicious device from the trusted list. The manager could also recognize potential attacks, inform the operator, and automatically warn other devices to increase monitoring or adapt security settings.

6. Discussion

6.1. Multi-Modal Architecture and Security Modes

Security modes are not a comprehensive solution to all security issues. Secure system design and development remain crucial for producing high-quality products and avoiding medical device recalls. Our middleware and framework aim not to make developers less concerned about security in system development. However, false-positive vulnerabilities or fake anomalies may trigger unnecessary mode switches and reduce functionality. So precautions must be taken to prevent the misuse of self-healing and self-protection mode mechanisms. An encrypted channel between the control unit (such as inventory) and medical devices is essential, along with authentication and monitoring to prevent and record unauthorized access. Manual mode switching by operators can also be problematic, and the "more than one eye" principle may be necessary for significant changes. Technical and organizational security measures are necessary to mitigate the risks of malicious insiders. The use of mode switching in IoMT devices can have both positive and negative effects. While it can increase complexity, increase maintenance, and lead to unintended behavior, it can also secure the system between vulnerability disclosure and patch installation, reduce the attack surface, make the system more resilient, and enable manual risk reduction even if the manufacturer has stopped service.

6.2. Mitigating Vulnerabilities by Switching Modes

Vulnerabilities are common for medical devices, and manufacturers often take a long time to develop and distribute patches. Over-the-air updates through a home station can speed up distribution, but this also poses additional risks, as demonstrated by update errors from other areas. If medical devices are not always connected, updates require hospital appointments. Mode switching can offer more flexibility and reduce the window of vulnerability. Our case study evaluation in (Riegler, Sametinger, Vierhauser, et al., 2023) found that mode switching between all modes improved security compared to using just a single web server. Mode switching resulted in fewer days at risk, lower CVSS score, more functionality, and flexibility to react to vulnerabilities. Mode switching can also be triggered by vulnerability scanners, intrusion detection systems, and early warning systems. The concept of mode switching can also be extended to react to suspicious data or security issues. Disabling critical healthcare system functions is always a last resort and challenging decision. Our architecture presents a solution to this problem by offering a range of modes instead of just on/off options. Using our architecture gives manufacturers more time to provide well-tested patches. Healthcare delivery organizations can use a central manager to communicate with medical devices, analyze their security status, alert patients, provide patches, and modify device settings as needed. However, preventing misuse of mode switching is important to avoid security breaches.

6.3. Self-Protection

If the IoMT device is not connected to a central authority like our proposed manager, the options for monitoring and control are restricted. To address this, a lightweight IDPS installed on the IoMT device can detect and prevent suspicious activities. However, incorrect or faulty detection can result in reduced functionality and usability. Another concern is related to the autonomy of the systems. In highly automated settings, critical events can go undetected amid the vast amount of data and an overreliance on the system. Therefore, automation should be introduced gradually, and log files utilized for post-mortem analysis. In Riegler, Sametinger, & Vierhauser (2023), we extended our previous work on multi-mode systems and present a distributed MAPE-K framework, which establishes a model for managing and controlling self-protective IoT devices. The framework was evaluated through a use case simulation of potential security issues, such as port scans and unauthorized login attempts, which confirmed that the framework can easily detect and mitigate diverse security threats.

7. Conclusion and Future Work

Securing medical devices is challenging due to their long lifespan, limited computing resources, and power management. Mode switching can overcome the time between a disclosed zero-day vulnerability and a security patch. Therefore, healthcare organizations and even patients get the capability to actively protect themselves by switching to a secure and safe mode with limited functionality but minimal attack surface and a limited range of activity for hackers. A resilient middleware and mode switching framework has been proposed to mitigate attacks and restrict attackers' activity.

Our research improves security in medical devices and makes them more robust against cyberattacks. We provide a mechanism for software safety and security risk mitigation in med-

ical devices, and we identified and developed methods of smart mode switching with an adaptive risk assessment. As a result, a medical device prototype/simulation show how smart mode switching can work. An evaluation with historical vulnerabilities showed the effectiveness of our new approach. Moreover, we give recommendations for medical device manufacturers and healthcare providers to ensure security. While security modes are not a complete solution and do not replace other precautions, simulations have shown they can reduce attack surfaces in case of vulnerabilities. However, their effectiveness against hardware trojans is limited, and more research is needed to determine their exact efficiency. By analyzing and correlating issues from multiple medical devices, anomalies and attacks can be detected and further attempts mitigated.

In the future, we aim to integrate modes deeper into software and combine them with software product lines to ward off targeted attacks. We plan to analyze our approach with existing systems and tools to enhance scalability and usability and to test its effectiveness. Additionally, we want to use machine learning for system behavior analysis and threat detection.

References

Abie, H., Savola, R., Bigham, J., Rotondi, D., & Da Bormida, G. (2010). Self-healing and secure adaptive messaging middleware for business-critical systems. *Int J Adv Secur*, *3*.

Ajagbe, S. A., Awotunde, J. B., Adesina, A. O., Achimugu, P., & Kumar, T. A. (2022). Internet of Medical Things (IoMT): Applications, Challenges, and Prospects in a Data-Driven Technology. In C. Chakraborty & M. R. Khosravi (Eds.), *Intelligent Healthcare: Infrastructure, Algorithms and Management* (pp. 299–319). Springer Nature. https://doi.org/10.1007/978-981-16-8150-9_14

Alemzadeh, H., Iyer, R. K., Kalbarczyk, Z., & Raman, J. (2013). Analysis of Safety-Critical Computer Failures in Medical Devices. *IEEE Security & Privacy*.

Almazyad, I., Rao, A., & Rozenblit, J. (2020). A Framework for Secure Data Management for Medical Devices. *2020 Spring Simulation Conference (SpringSim)*, 1–12. https://doi.org/10.22360/SpringSim.2020.MSM.005

Alsubaei, F., Abuhussein, A., & Shiva, S. (2017). Security and Privacy in the Internet of Medical Things: Taxonomy and Risk Assessment. *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 112–120. https://doi.org/10.1109/LCN.Workshops.2017.72

Arcaini, P., Riccobene, E., & Scandurra, P. (2015). Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 13–23. https://doi.org/10.1109/SEAMS.2015.10

Barrett, M. P. (2018). Framework for Improving Critical Infrastructure Cybersecurity Version 1.1. *NIST*. https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11

Bochniewicz, E., Chase, M. P., Coley, S. M. C., Wallace, K., Weir, M., & Zuk, M. (2021). *Playbook for Threat Modeling Medical Devices*. https://www.mitre.org/publications/technical-papers/playbook-threat-modeling-medical-devices

BSI. (2018). *Cyber Security Requirements for Network-Connected Medical Devices*. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/ICS/Medical_Devices_CSE_132.html

Burns, A. J., Johnson, M. E., & Honeyman, P. (2016). A brief chronology of medical device security. *Communications of the ACM*, *59*(10), 66–72. https://doi.org/10.1145/2890488

Carreon, N. A., Lu, S., & Lysecky, R. (2021). Probabilistic Estimation of Threat Intrusion in Embedded Systems for Runtime Detection. *ACM Transactions on Embedded Computing Systems*, *20*(2), 14:1-14:27. https://doi.org/10.1145/3432590

Carreon-Rascon, A. S., & Rozenblit, J. W. (2022). Towards Requirements for Self-Healing as a Means of Mitigating Cyber-Intrusions in Medical Devices. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1500–1505. https://doi.org/10.1109/SMC53654.2022.9945507

Chen, G., & Kotz, D. (2000). *A Survey of Context-Aware Mobile Computing Research* (Computer Science Technical Report No. TR2000-381). Dartmouth College. http://cs.dartmouth.edu/~dfk/papers/chen:survey-tr.pdf

Chen, T., & Phan, L. T. X. (2018). SafeMC: A System for the Design and Evaluation of Mode-Change Protocols. *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 105–116. https://doi.org/10.1109/RTAS.2018.00021

CISA. (2022). *ICS Medical Advisory (ICSMA-21-187-01) Philips Vue PACS (Update A)*. https://www.cisa.gov/uscert/ics/advisories/icsma-21-187-01

Claroty. (2022). *State of XIoT Security Report*. https://claroty.com/press-releases/iot-vulnerability-disclosures-grew-57-percent-from-2h21-to-1h22

de Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N. M., Vogel, T., Weyns, D., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Engels, G., … Wuttke, J. (2013). Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In R. de Lemos, H. Giese, H. A. Müller, & M. Shaw (Eds.), *Software Engineering for Self-Adaptive Systems II* (Vol. 7475, pp. 1–32). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35813-5_1

Easttom, C., & Mei, N. (2019). Mitigating Implanted Medical Device Cybersecurity Risks. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 0145–0148. https://doi.org/10.1109/UEMCON47517.2019.8992922

Elhoseny, M., Thilakarathne, N. N., Alghamdi, M. I., Mahendran, R. K., Gardezi, A. A., Weerasinghe, H., & Welhenge, A. (2021). Security and Privacy Issues in Medical Internet of Things: Overview, Countermeasures, Challenges and Future Directions. *Sustainability*, *13*(21), Article 21. https://doi.org/10.3390/su132111645

Espinoza, J. R., Padilla, V. S., & Velasquez, W. (2017). IoT Generic Architecture Proposal Applied to Emergency Cases for Implanted Wireless Medical Devices. *Hong Kong*, *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*.

FDA. (2016). *Postmarket Management of Cybersecurity in Medical Devices—Guidance for Industry and Food and Drug Administration Staff*. http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM482022.pdf

FDA. (2017). *Firmware Update to Address Cybersecurity Vulnerabilities Identified in Abbott's (formerly St. Jude Medical's) Implantable Cardiac Pacemakers*.

http://www.fda.gov/medical-devices/safety-communications/firmware-update-address-cybersecurity-vulnerabilities-identified-abbotts-formerly-st-jude-medicals

FDA. (2022). *Content of Premarket Submissions for Management of Cybersecurity in Medical Devices—Draft Guidance for Industry and Food and Drug Administration Staff*. https://www.fda.gov/media/119933/download

Firesmith, D. (2019, November 25). *System Resilience: What Exactly is it?* https://insights.sei.cmu.edu/sei_blog/2019/11/system-resilience-what-exactly-is-it.html

Forbes. (2019, July 29). *Critical "Update Now" Warning Issued For VxWorks OS Inside 2 Billion IoT Devices*. Forbes. https://www.forbes.com/sites/zakdoffman/2019/07/29/warning-as-2-billion-medical-industrial-and-enterprise-iot-devices-at-risk-of-attack/

Garbelini, M. E., Chattopadhyay, S., & Wang, C. (2020). SweynTooth: Unleashing Mayhem over Bluetooth Low Energy. *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*. USENIX ATC'20.

Gartner. (2021). *Gartner Predicts By 2025 Cyber Attackers Will Have Weaponized Operational Technology Environments to Successfully Harm or Kill Humans*. Gartner. https://www.gartner.com/en/newsroom/press-releases/2021-07-21-gartner-predicts-by-2025-cyber-attackers-will-have-we

Halperin, D., Benjamin, T. S. H., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., & Maisel, W. H. (2008). Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. *Proceedings of the 2008 IEEE Symposium on Security and Privacy*.

IMDRF. (2020). *Principles and Practices for Medical Device Cybersecurity*. International Medical Device Regulators Forum (IMDRF). https://www.imdrf.org/sites/default/files/docs/imdrf/final/technical/imdrf-tech-200318-pp-mdc-n60.pdf

Joint Task Force Transformation Initiative. (2014). *Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans* (NIST Special Publication (SP) 800-53A Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53Ar4

Kagita, M. K., Thilakarathne, N., Gadekallu, T. R., & Maddikunta, P. K. R. (2022). A Review on Security and Privacy of Internet of Medical Things. In U. Ghosh, C. Chakraborty, L. Garg, & G. Srivastava (Eds.), *Intelligent Internet of Things for Healthcare and Industry* (pp. 171–187). Springer International Publishing. https://doi.org/10.1007/978-3-030-81473-1_8

Kaplan, D. (2011). Insulin pumps can be hacked. *SC Magazine*. http://www.scmagazine.com/black-hat-insulin-pumps-can-be-hacked/article/209106/

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41–50. https://doi.org/10.1109/MC.2003.1160055

Li, C., Raghunathan, A., & Jha, N. K. (2011). Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. *IEEE 13th International Conference on E-Health Networking, Applications and Services*, 150–156. https://doi.org/10.1109/HEALTH.2011.6026732

Lu, S., & Lysecky, R. (2019). Data-driven Anomaly Detection with Timing Features for Embedded Systems. *ACM Transactions on Design Automation of Electronic Systems*, *24*(3), 1–27. https://doi.org/10.1145/3279949

Lu, S., Seo, M., & Lysecky, R. (2015). Timing-Based Anomaly Detection in Embedded Systems. *20th Asia and South Pacific Design Automation Conference*, 809–814.

MDCG. (2019). *Guidance on Cybersecurity for medical devices*. Medical Device Coordination Group (MDCG), European Commission. https://ec.europa.eu/docsroom/documents/41863/attachments/1/translations/en/renditions/native

Medcrypt. (2022). *What the medical device industry can learn from past cybersecurity vulnerability disclosures*. https://www.medcrypt.co/whitepaper_resources/MedCrypt_Vuln_Disclosures_2022.pdf

Medtronic. (2022a, August). *The MiniMedTM 630G and 770G Insulin Pumps*. https://www.medtronic.com/us-en/healthcare-professionals/therapies-procedures/diabetes/education/diabetes-digest/minimed-insulin-pumps.html

Medtronic. (2022b, September 20). *Urgent Medical Device Correction MiniMed™ 600 Series Pump System Communication Issue*. Medtronic Diabetes. https://www.medtronicdiabetes.com/customer-support/product-and-service-updates/notice19-letter

Palo Alto Networks. (2022, March 2). Infusion Pump Vulnerabilities: Common Security Gaps. *Unit42*. https://unit42.paloaltonetworks.com/infusion-pump-vulnerabilities/

Paul, N., Kohno, T., & Klonoff, D. C. (2011). A Review of the Security of Insulin Pump Infusion Systems. *Journal of Diabetes Science and Technology*, *5*(6).

Pisani, F., de Oliveira, F. M. C., Gama, E. S., Immich, R., Bittencourt, L. F., & Borin, E. (2020). *Fog Computing on Constrained Devices: Paving the Way for the Future IoT*. https://doi.org/10.3233/APC200003

Rao, A., Carreón, N. A., Lysecky, R., & Rozenblit, J. (2022). FIRE: A Finely Integrated Risk Evaluation Methodology for Life-Critical Embedded Systems. *Information*, *13*(10), Article 10. https://doi.org/10.3390/info13100487

Rao, A., Carreón, N., Lysecky, R., Rozenblit, J. W., & Sametinger, J. (2019). Resilient Security of Medical Cyber-Physical Systems. *Database and Expert Systems Applications - DEXA 2019 International Workshops BIOKDD, IWCFS, MLKgraphs and TIR, Linz, Austria, August 26-29, 2019, Proceedings*, 95–100. https://doi.org/10.1007/978-3-030-27684-3_13

Reports And Data. (2021). *Market value of the internet of medical things worldwide in 2019 and 2027*. Statista. https://www.statista.com/statistics/1264333/global-iot-in-healthcare-market-size/

Riegler, M., & Sametinger, J. (2020). Mode Switching from a Security Perspective: First Findings of a Systematic Literature Review. In G. Kotsis, A. M. Tjoa, I. Khalil, L. Fischer, B. Moser, A. Mashkoor, J. Sametinger, A. Fensel, & J. Martinez-Gil (Eds.), *Database and Expert Systems Applications* (pp. 63–73). Springer International Publishing. https://doi.org/10.1007/978-3-030-59028-4_6

Riegler, M., Sametinger, J., & Rozenblit, J. W. (2022). Context-Aware Security Modes For Medical Devices. *2022 Annual Modeling and Simulation Conference (ANNSIM)*, 372–382. https://doi.org/10.23919/ANNSIM55834.2022.9859283

Riegler, M., Sametinger, J., & Rozenblit, J. W. (2023). *Self-protective Healthcare Systems with Medical IoT Devices*. submitted for publication.

Riegler, M., Sametinger, J., & Vierhauser, M. (2023). *A Distributed MAPE-K Framework for Self-Protective IoT Devices*. submitted for publication.

Riegler, M., Sametinger, J., Vierhauser, M., & Wimmer, M. (2023). A model-based mode-switching framework based on security vulnerability scores. *Journal of Systems and Software*, 111633. https://doi.org/10.1016/j.jss.2023.111633

Ross, R., McEvilley, M., & Carrier Oren, J. (2016). *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems* (NIST SP 800-160; p. NIST SP 800-160). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160

Sametinger, J., & Rozenblit, J. (2016). Security Scores for Medical Devices. *9th International Conference on Health Informatics*, 533–541. https://doi.org/10.5220/0005838805330541

Sametinger, J., Rozenblit, J., Lysecky, R., & Ott, P. (2015, April). Security Challenges for Medical Devices. *Communications of the ACM*, *58*(4), 74–82. https://doi.org/10.1145/2667218

Sametinger, J., & Steinwender, C. (2017). Resilient Context-Aware Medical Device Security. *International Conference on Computational Science and Computational Intelligence, Symposium on Health Informatics and Medical Systems (CSCI-ISHI)*, 1775–1778. https://doi.org/10.1109/CSCI.2017.310

Samie, F., Tsoutsouras, V., Bauer, L., Xydis, S., Soudris, D., & Henkel, J. (2016). Computation offloading and resource allocation for low-power IoT edge devices. *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 7–12. https://doi.org/10.1109/WF-IoT.2016.7845499

Samie, F., Tsoutsouras, V., Bauer, L., Xydis, S., Soudris, D., & Henkel, J. (2019). Oops: Optimizing Operation-mode Selection for IoT Edge Devices. *ACM Transactions on Internet Technology*, *19*(2), 22:1-22:21. https://doi.org/10.1145/3230642

Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S., & Wang, G. (2018). Security and Privacy in the Medical Internet of Things: A Review. *Security and Communication Networks*, *2018*, 9.

Suresh, V., Ramson, J., & Jegan, D. (2020). *Internet of Medical Things (IoMT)—An overview* (p. 104). https://doi.org/10.1109/ICDCS48716.2020.243558

Thomasian, N. M., & Adashi, E. Y. (2021). Cybersecurity in the Internet of Medical Things. *Health Policy and Technology*, *10*(3), 100549. https://doi.org/10.1016/j.hlpt.2021.100549

Tomić, I., Chen, P.-Y., Breza, M. J., & McCann, J. A. (2018). Antilizer: Run Time Self-Healing Security for Wireless Sensor Networks. *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 107–116. https://doi.org/10.1145/3286978.3287029

Trusted Computing Group (TCG). (2022, March 11). *Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.59*. https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM2_r1p59_Part1_Architecture_pub.pdf

U.S. Federal Bureau of Investigation (FBI). (2022, September 12). *Industry Alert: Unpatched and Outdated Medical Devices Provide Cyber Attack Opportunities*. https://www.ic3.gov/Media/News/2022/220912.pdf

US Nuclear Regulatory Commission. (1995). *Standard technical specifications: Babcock and Wilcox Plants. Revision 1* (NUREG--1430-Vol.1-Rev.1, 87064; p. NUREG--1430-Vol.1-Rev.1, 87064). https://doi.org/10.2172/87064

van Landuyt, D., Pasquale, L., Sion, L., & Joosen, W. (2021). Threat modeling at run time: The case for reflective and adaptive threat management (NIER track). *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 203–209. https://doi.org/10.1109/SEAMS51251.2021.00034

Volterrani, M., & Sposato, B. (2019). Remote monitoring and telemedicine. *European Heart Journal Supplements: Journal of the European Society of Cardiology*, *21*(Suppl M), M54–M56. https://doi.org/10.1093/eurheartj/suz266

Yaacoub, J.-P. A., Noura, M., Noura, H. N., Salman, O., Yaacoub, E., Couturier, R., & Chehab, A. (2020). Securing internet of medical things systems: Limitations, issues and recommendations. *Future Generation Computer Systems*, *105*, 581–606. https://doi.org/10.1016/j.future.2019.12.028

Yaqoob, T., Abbas, H., & Atiquzzaman, M. (2019). Security Vulnerabilities, Attacks, Countermeasures, and Regulations of Networked Medical Devices—A Review. *IEEE Communications Surveys Tutorials*, *21*(4), 3723–3768. https://doi.org/10.1109/COMST.2019.2914094