CONTEXT-AWARE SECURITY MODES FOR MEDICAL DEVICES

Michael Riegler
Johannes Sametinger

LIT Secure and Correct Systems Lab
and Dept. of Business Informatics
Johannes Kepler University Linz
Altenberger Straße 69, Linz, AUSTRIA
{michael.riegler, johannes.sametinger}@jku.at

Jerzy W. Rozenblit

Dept. of Electrical and Computer Engineering
and Dept. of Surgery
University of Arizona
E Speedway Blvd, Tucson, AZ, USA
jerzyr@arizona.edu

ABSTRACT

Medical devices require the provision of life-critical functionality even under adverse conditions. We imagine to model (at design time) and to switch (at run-time) security modes in a self-adaptive way, thus, reducing attack surfaces in case of a malfunction, attack, or when vulnerabilities become known. Modes return back to normal when patches are provided and installed. Context-aware devices can resiliently provide a degraded mode of operation with a smaller attack surface instead of completely disabling the whole system or a device recall. Healthcare organizations and patients should actively protect themselves by implicitly or explicitly switching to modes with limited activity ranges for attackers. We use simulation to check all circumstances and the self-healing functionality to return to normal mode. In this paper, we present our ongoing work to make medical devices more secure. We discuss how modes can support that, how they are defined, and what challenges they provide.

Keywords: medical devices, security modes, context-awareness, resilience

1 INTRODUCTION

Medical Devices (MDs) support human well-being and save lives. COVID-19 has pushed the usage of telehealth, wearable devices, home monitoring systems and others. These devices' increasing interoperability poses threats to the health of patients and even their lives. Implantable devices like pacemakers or insulin pumps are especially critical, as they can directly influence patients' conditions. Remote monitoring can increase patients' quality of life. Nevertheless, unsecured MDs potentially pose threats to millions of patients worldwide. According to Claroty (2022), the vulnerability disclosures increased by 110% in the last four years. Recent reports revealed that 75% of more than 200,000 infusion pumps have had security gaps (Palo Alto Networks 2022). These can lead to privacy issues, battery depletion, malfunction up to death threats, extortion, or remote assassination with manipulated settings and deadly doses of medications. Some MDs are using off-the-shelf (OTS) hardware and software. Thus, vulnerabilities like URGENT/11, SweynTooth, Ripple20, AMNESIA:33, BadAlloc, Nucleus:13, Log4Shell, and others can affect them. The *U.S. Food and Drug Administration* (FDA) has issued several recalls of MDs in the last years because of potential cybersecurity risks. Fortunately, no incidents have become known so far. However, Gartner predicts that by 2025 "cyber attackers will have weaponized operational technology environments to successfully harm or kill humans" (Gartner 2021). Securing these devices provides many challenges, including hardware and

software, but also organizational and regulatory challenges, see (Sametinger et al. 2015). Due to the long lifetime of MDs and their limited update functionality, it is conceivable that quantum computers may be able to crack current encryption algorithms in the future. Understandably enough, stopping the operation of MDs in case of a threat is not the desired option. A restart is also undesirable, especially if it can possibly lead to death (Medtronic 2021). Nevertheless, *Health Care Delivery Organizations* (HDOs) and affected patients should not have to wait for updates or recalls of the *Medical Device Manufacturers* (MDMs). Because in the meantime, attackers can also find out about security gaps and exploit them.

In this paper, we propose the design of context-aware security modes to resiliently protect MDs by switching them manually or automatically depending on whether anomalies occur or vulnerabilities become known. In Section 2, we discuss related work and describe common methods to secure MDs in Section 3. MDs levels of concern follow in Section 4. In Section 5, we present our considerations about the use of security modes for MDs. Modeling and simulation issues follow in Section 6. Finally, we draw our conclusions in Section 8.

2 RELATED WORK

Considerable work has been done in the analysis of modes in MDs. Usually fail-safe modes or states provide proper operation in case of an adverse event, such as an update failure or other hardware and software failures. We understand modes more as a logical human observable framework that combines system states, provides specific functionalities, and faces different security risks. Multi-mode systems are well known in other domains like aviation, automotive, and energy (SmartCockpit 2004, Autosar 2017, NRC 2019), where modes are used to divide and manage complexity, have specific configurations, and consist of specific behaviors. For example, 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. Each mode provides a set of functionalities, and some actions are prohibited for safety reasons, like thrust reversal during take-off and flying. First findings of mode switching from a security perspective are provided in our systematic literature review (Riegler and Sametinger 2020).

BSI (2018) have introduced modes for medical operation, device configuration, and technical maintenance and evaluated the patient risk associated with a specific vulnerability with regard to modes. Ross et al. (2016) suggest operational, contingency, degraded, and alternative modes of operation as a response to disruptions, hazards, and other threats that may occur. Rao et al. (2018) 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, e.g., a power-saving mode to avoid power outages or a mode with lower sampling rates and work offload in case of low battery (Alemzadeh et al. 2013, Samie et al. 2019). Easttom and Mei (2019) propose a software shim with a normal and an emergency mode to protect implanted medical devices. If an anomaly is detected, the device switches to the emergency mode and will only allow data synchronization.

3 SECURING MEDICAL DEVICES

MDMs are responsible to ensure the quality of their products, which includes addressing cybersecurity risks. *Failure Mode and Effect Analysis* (FMEA) is a common procedure to analyze system reliability and safety. Security has to be considered throughout the whole life-cycle of a product, beginning from design, development, and distribution through to maintenance and decommissioning. The FDA provides pre- and postmarket guidance to build secure devices by design and reduce the risk to patient health. A *Bill of Materials* (BOM) provides an overview of hardware and software components like which commercial, open-source, and/or OTS software is used in a product. BOMs make monitoring more manageable and reduce the response time if problems or vulnerabilities in the supply chain occur (FDA 2016b, FDA 2018).

According to the NIST Cybersecurity Framework, MDMs should consider functionality to identify, protect, detect, respond and recover in order to prevent unauthorized use, loss of confidentiality, integrity, availability, and patient harm (Barrett 2018). Assets, threats, their likelihood, and the possible impacts need to be identified and risks calculated. MITRE provides a playbook to identify threats for MDs and describes how to mitigate them (Bochniewicz et al. 2021). We need safeguards and other protection methods and mechanisms to limit access and ensure that safety-critical commands can be executed only from trusted and authorized users and devices. If vulnerabilities of a MD's OTS software become known or if there is any kind of anomaly, we have to take action before exploits (can) happen. Even in case of failure, misconfiguration, or possible attack, MDs should be able to reduce potential negative impact and resiliently recover from this situation. Different usernames and passwords per MD are not enough. Default and weak passwords may be revealed by reverse engineering or guessed with brute-force attacks. There exists considerable work about different authentication protocols based on proxies, biometrics, proximity, and trusted third parties.

Based on the chronology of medical device security, "threats and zero-day vulnerabilities must be addressed over the entire useful life of medical devices" (Burns et al. 2016). The development and the distribution of updates to fix them needs to be accelerated. Even if MDMs discontinue the support of devices or even if they go bankrupt, we should not end up without options against potential threats. There are many reasons why most MDs cannot be updated and patched like regular computers. Common operating systems and other platforms are replaced every couple of years. However, some MDs may have a lifespan from 15 to 20 years, need legacy systems, and often do not provide automatic update mechanisms as most of them are not permanently connected to the HDO or the manufacturer. Automatic updates can even increase the devices' attack surfaces in case they introduce new vulnerabilities, backdoors, or misconfigurations. Therefore, updates must be delivered from trusted and verified sources. In addition, updates for several devices must not require a reboot, e.g., in most implanted devices like cardiac pacemakers. Regulations may also hinder MDMs and HDOs from quickly rolling out updates and patches. As long as vulnerabilities are not fixed, attackers can find that out and exploit them. (Sametinger and Rozenblit 2016)

4 LEVELS OF CONCERN

The FDA has assigned device types to the regulatory classes I, II, or III, which are based on the level of control needed to assure the safety and effectiveness of a device. The higher a device's risk, the higher its class (FDA 2016a). The FDA has also introduced a level of concern for MDs. It is a measure referring to "an estimate of the severity of injury that a device could permit or inflict, either directly or indirectly, on a patient or operator as a result of device failures, design flaws, or simply by virtue of employing the device for its intended use" (FDA 2005).

Privacy and safety concerns for MDs have been introduced in (Sametinger and Steinwender 2017). They have used sensitivity, impact, and exposure of devices to define privacy and safety concerns. Exposure combined with sensitivity yields the degree of exposure of sensitive information. Devices that do not store or process sensitive information do not pose a privacy threat. The same is true if a device is not exposed. Safety is not an issue with devices without an impact on patients. The same is true if the device is not exposed, even if it does have an impact. Security is not a problem when devices are not exposed to their environment. Privacy is at stake if a device stores sensitive information and we can access it from outside the device. Safety is at stake if a device impacts patients, and we can control this impact from outside the device. This can happen if the security is weak. For example, if attackers get access to pacemakers and change the clock rate or trigger emergency shocks, they harm patients and, thus, make a security issue also to a safety issue. (Sametinger and Steinwender 2017).

In 2018, the FDA replaced the levels of concern with a two-tiered risk approach (FDA 2018). Wired or wireless connected MDs, which can harm one or multiple patients, are considered to have a higher cyberse-

curity risk and higher requirements for approval. Devices in this category include implantable cardioverters, defibrillators (ICDs), pacemakers, left ventricular assist devices (LVADs), brain and neuro stimulators, dialysis devices, infusion and insulin pumps, home monitors and programmers. Physical hardware attacks on implanted MD are difficult to imagine without the patient noticing. Therefore, these devices are mostly affected by attacks on wireless connections. In contrast, wearables, home monitors, and others must also handle this kind of threat. In addition, manipulated hardware or firmware could find its way into MDs via the supply chain.

5 SECURITY MODES FOR MEDICAL DEVICES

MDs can be secure or insecure. That depends on the existence of vulnerabilities, their severeness, and whether they are known. This is independent of a device's sensitivity, impact, or exposure. Security scores introduced in (Sametinger and Steinwender 2017) classify MDs from a privacy and safety perspective. We suggest a defense-in-depth strategy with security modes for these devices. For example, when vulnerabilities become known or when devices are exploited, they can be made more secure by reducing their exposure. In this scenario, the MDM, the HDO, or even the patients would have to take steps to put a device in a different mode with changed security scores. These modes will allow them to reduce the attack surface and attackers' range of activity. If devices have enough processing power (and power supply is adequate), they can also actively react to their environment. For example, they can switch to a more secure mode when detecting any potential intrusion.

Context awareness manifests itself in various forms (Chen and Kotz 2000). For example, active context-awareness means automatically adapting the behavior according to a discovered context. MDs can react differently depending on whether there is a connection, e.g., to a home station or the hospital. In case of an emergency, the device will behave differently. Passive context-awareness means that a device manufacturer may inform device users about alerts for a specific brand or model. HDO and patients can then take appropriate action and start operating that brand or model in a different mode. An MD may also directly contact a server to get information about its security context. Unfortunately, resource limitations like battery power may inhibit such a course of action. However, home monitoring systems can provide such a service and propagate the information to an implanted device. Another option is that a patient confirms access with a button on the home station if the HDO requests access to a MD. In addition, the MD or the home station can notify the HDO and/or the MDM about mode switches. The best protection can be provided by active context-awareness, e.g., by anomaly detection, where a device itself can recognize abnormal behavior and initiate countermeasures like switching to a more secure mode (Lu et al. 2015, Lu and Lysecky 2019, Carreon et al. 2021).

When a vulnerability becomes known, the window of exposure opens and will be closed only when a patch is provided and installed. The goal is to keep windows of exposure short. But it is up to device manufacturers. Unfortunately, patients and HDOs cannot do much about it. When a device can actively or passively switch to a more secure mode, then it will be less vulnerable during the window of exposure, providing resilience of life-critical functionality. When the time of exposure closes, a device can (actively or passively) switch back to normal, i.e., recover and provide full functionality again. This kind of self-healing process must be carefully implemented so that it cannot be abused by itself. For example, the *Trusted Platform Modules* (TPMs) increases the healing time after each failed attempt (TCG 2019).

6 MODE MODELING AND MODE SIMULATION

As MDs hardware and software may not be readily accessible to researchers, we intend to model and simulate a specific device. This will require that we model at least a subset of modes for a specific device, followed by a simulated run with real-world vulnerabilities. We want to simulate ways of the self-adaptive

restriction provided by the different modes and the self-healing process. We have demonstrated the usefulness of this approach for web applications, where we had simulation running over a time-span of two simulated years, see (Riegler et al. 2022). We had retrieved the *Common Vulnerability Exposures* (CVEs) as well as their severity scores based on the *Common Vulnerability Scoring System* (CVSS), and provided patches, based on a BOM with the information of all installed software versions. We used this information as the foundation for hypothetical mode switch decisions. In fact, we used a *Mode Domain-Specific Language* (MDSL) to define a multi-modal architecture and used it to protect systems resiliently. We had defined different modes by running different software versions from different manufacturers and, thus, with different vulnerabilities. If vulnerabilities become known for a currently running software, switching to another version can mitigate this vulnerability. In our MDSL, excerpt see Figure 1, each mode can have specific start and stop actions, conditions, and a priority. Each mode uses specific software versions, which can be affected by vulnerabilities and lead to a mode switch. If there are equal risks for two or more modes, the priority is used for the decision. For example, Figure 2 shows the mode definition for ApacheWithPhp.

```
'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 Defintion

```
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 Defintion ApacheWithPhp

We have used historic vulnerabilities and patches of two years to compare risk scores and show that mode switching can reduce the windows of exposure. In our sample scenario, we could reduce the windows of exposure with 7 to 11 mode switches from 536 to 8 days, resulting in zero known risk in 98.9% of the analyzed time. Figure 3 shows summed up scores of vulnerabilities over a time-span of two years and compares the modes ApacheWithPhp, NginxWithPhp and Mode-Switching (Riegler et al. 2022).

We plan to use the MDSL for a cardiac pacemaker scenario and extend it with rules and events to provide more specific mode-switching settings. We also want to monitor and control several multi-mode scenarios and investigate how to best increase the security and resiliency of MDs. We imagine triggering mode switches from outside, as pacemakers have too little computing power for threat detection. Usually, there

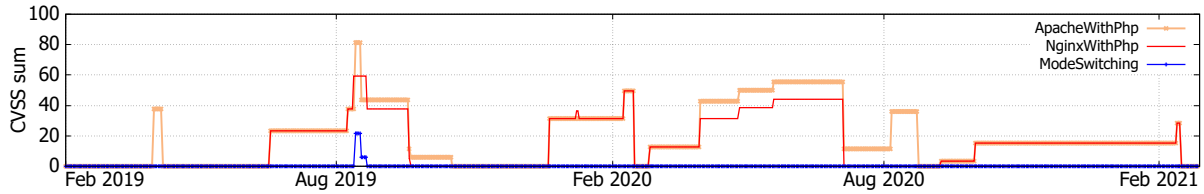

Figure 3: Vulnerability Scores of Web Server Scenario

is a six-monthly check-up at the HDO, where the doctor checks the log for abnormalities, the state of the battery and optimizes the programming if necessary. These check-ups can also be used to trigger security mode switches if needed. Alternatively, home stations can initiate such switches more timely.

6.1 Proposal of a Multi-Modal Architecture

Figure 4 provides an overview of our proposed multi-modal architecture. The three main parts of the architecture are the *configuration part* with our MDSL, the *mode control part* within the MD, and the *inventory part*. As a first step, we define (1) the desired modes using a *System Mode Description* in our MDSL and save it in the inventory. It contains information about the mode behavior and the actions to be taken if we want to switch modes. Then we automatically can generate (2) the *System Mode Configuration* based on the *System Mode Description*. The *Mode Control* component runs the system configuration, analyzes events, and eventually executes a mode switch. These can be triggered by a *Log File Analyzer* (11), an *Intrusion Detection System* (IDS) (10), or a *Vulnerability Analyzer* (5). The *Vulnerability Manager* automatically collects (3) CVEs, patches, and exploits from public databases and vendors for the parts of the systems as specified in the *System Mode Description*. Information like new CVEs, patches, exploits, or changed CVSS scores are forwarded (4) to the *Event-Analyzer*. We sum up the CVSS scores of all software components as well as modes, and use them for prioritization. The scores for each mode are used to determine whether a mode switch is necessary. While new CVEs and exploits increase total CVSS scores, patches decrease them, assuming they have been installed. Depending on rules and policies, some events (5) will be automatically sent to the MD, while others may need the human-in-the-loop for further investigation. Thresholds are useful to prevent frequent mode changes triggered by intrusion detection systems or by log file analyzers. The *Event Analyzer*, which is part of the MD, receives events and decides whether the current mode is suitable. If needed the *Mode Switcher* is called (6) to perform the specific tasks to start the new mode (7). For safety reasons and to reduce the attack surface, manual mode switches, triggered by the operator (8), can be necessary for a specific MD or a group of MDs. The operator manages (9) the inventory of several MDs and the settings of the *Vulnerability Manager*.

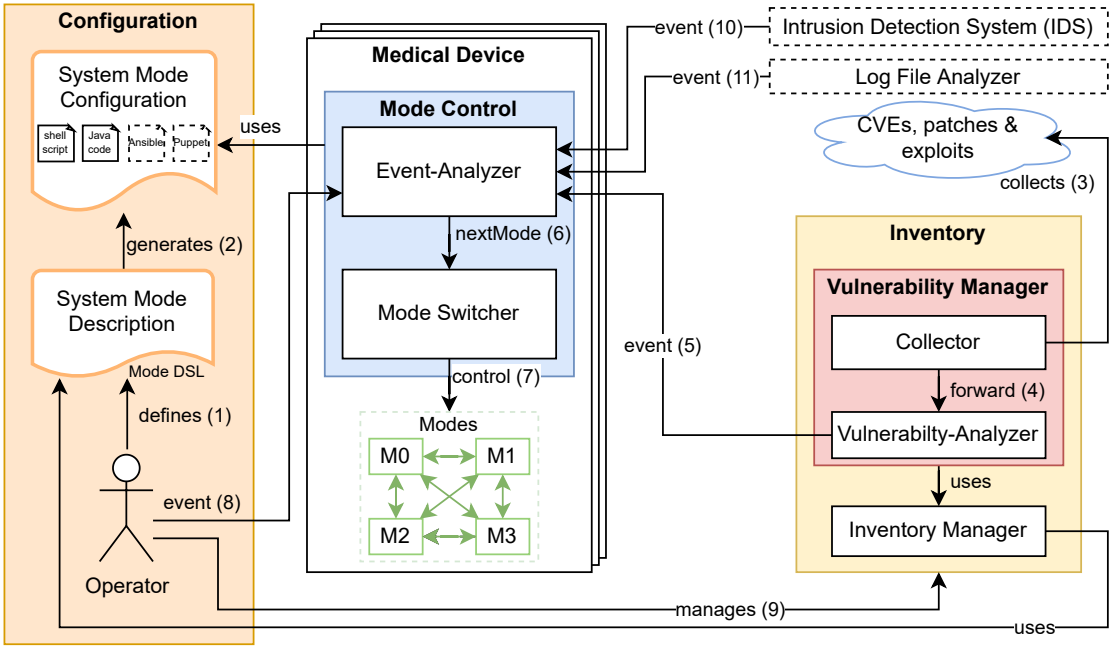

Figure 4: Conceptual Overview of our Multi-modal Architecture

6.2 Sample Mode Simulation

There are not many CVE entries available for MDs. But the small number of publicly known security issues can be deceptive, as manufacturers may refrain from making vulnerabilities public. According to Medcrypt (2022), *ICS-CERT medical advisories* (ICSMAs) have increased by 490% since the FDA released their Postmarket Cybersecurity Guidance in 2016. Most advisories combine multiple CVEs, with user-authentication mismanagement and code defects as the most common root causes. In 71% (87) of the 122 advisories, researchers were involved in discovering the 355 found vulnerabilities. Of 15 device types, infusion pumps, imaging software, patient monitors, and cardiac rhythm management were the most affected and had nearly 50% of the total found vulnerabilities. Some MDMs provide detailed information only for their registered customers or keep incidents confidential. Therefore it is hard to reflect when a vulnerability was discovered, shared with the vendor, publicly disclosed, and when an update was provided.

We had a closer look at vulnerabilities of MDs and picked the medical advisory ICSMA-21-187-01 for the *Philips Vue Picture Archiving and Communication System (PACS)* (CISA 2022). We have chosen this example, because it is from this year (2022) and contained a total of 11 vulnerabilities in contrast to many others where only one vulnerability was involved in most cases. Figure 5 shows the timeline and the sum of all vulnerability scores with and without mode switching. Code defects and insecure third-party libraries were the root cause of most vulnerabilities. It took over nine years to fix the publicly known CVE-2012-1708. Even four third-party vulnerabilities with public known exploits were patched only up to four years later. On average, it took more than two years to provide a fix. Our suggested multi-modal system can detect third-party vulnerabilities and provide a method to mitigate the situation until a patch is provided. Usually, several programs are available to solve a specific task. Instead of using only one type of software like *7-Zip* or *Redis*, we can use alternative implementations like *gzip* or another key-value database. We can also transfer data via HTTPS instead of using the *Apache JServ Protocol*. The use of additional implementations and protocols makes systems more complex but allows flexibility during operation. In the current example, mode switching between third-party software can reduce the CVSS sum on 1404 of 3760 days. The average CVSS sum decreases by 41% from 28.2 to 16.6. We are not limited to switch between implementations only, but can also switch modes between different configurations of single implementations.

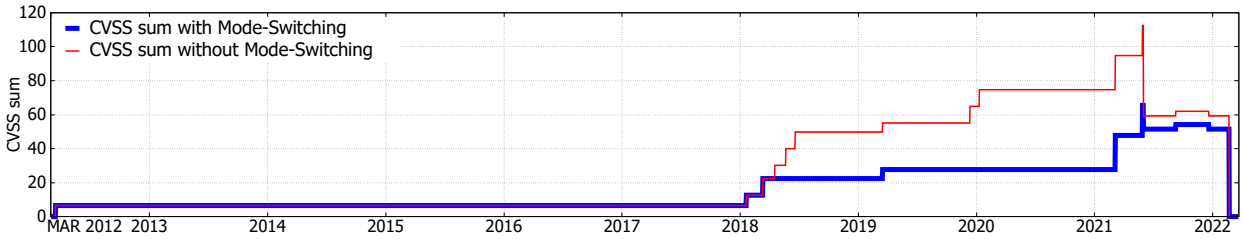

Figure 5: Vulnerability Scores of Philips Vue PACS ICSMA-21-187-01

7 DISCUSSION

Vulnerabilities in MDs are the rule rather than the exception. Another rule is it takes a long time until MDMs develop and distribute patches. Over-the-air updates, for example, through a home station, can accelerate distribution, but impose additional risks, as update errors from other areas have shown. If MDs are not (permanently) connected, hospital appointments are needed to install updates. Mode switching can provide more flexibility and shorten the window of exposure. At the same time, MDMs have more time to provide extensively tested patches.

Security modes will not solve all security issues. Secure system design and development are still important to provide high-quality products and mitigate MD recalls. It is not the goal to make developers less concerned about secure system development. In addition, false-positive vulnerabilities or fake anomalies can lead to unnecessary mode switches and reduced functionality. Precautions have to be taken to avoid that the self-healing mechanism of modes cannot be misused. It is essential to have an encrypted channel between the control unit, e.g., the inventory, and the MDs. Furthermore, authentication and monitoring are needed to prohibit and document illegal access. Manual mode switching by operators can also cause issues. A principle of more than one eye may be necessary for serious changes.

8 CONCLUSION

Securing MDs is not an easy task. MDs have a lifetime up to 20 years and often provide only limited computing resources, and have to carefully manage the power and battery utilization. We have presented current work on modeling and an architecture for a mode-switching framework to protect MDs resiliently. Simulations have shown that predefined modes can help mitigate attacks and restrict attackers' range of activity in case of vulnerabilities. Based on the context MDs can switch their mode and reduce their attack surface or increase functionality.

Security modes are not a silver bullet and do not replace other security defenses and precautions. The level of efficiency is yet to be determined exactly. For example, security modes will have limited effectiveness against hardware trojans. However, security modes provide an additional effective defense against many vulnerabilities that may affect MDs, by temporarily limiting attack surfaces if needed.

ACKNOWLEDGMENTS

This work has partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria, the Austrian Marshall Plan Foundation, and the National Science Foundation under Grant Number 1622589 "Time-Centric Modeling of Correct Behaviors for Efficient Non-intrusive Runtime Detection of Unauthorized System Actions." 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.

REFERENCES

Alemzadeh, H., R. K. Iyer, Z. Kalbarczyk, and J. Raman. 2013, Jul. "Analysis of Safety-Critical Computer Failures in Medical Devices". *IEEE Security Privacy* vol. 11 (4), pp. 14–26.

Almazyad, I., A. Rao, and J. Rozenblit. 2020, May. "A Framework for Secure Data Management for Medical Devices". In *SpringSim 2020*, pp. 1–12. 2020 Spring Simulation Conference. Fairfax, VA, USA.

Autosar 2017. "Guide to Mode Management". https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_EXP_ModeManagementGuide.pdf. Accessed Mar. 14, 2022.

Barrett, M. 2018. *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*. Number NIST SP 800-160. National Institute of Standards and Technology (NIST).

Bochniewicz, E., M. P. Chase, S. M. C. Coley, K. Wallace, M. Weir, and M. Zuk. 2021, Nov. *Playbook for Threat Modeling Medical Devices*. MITRE and the Medical Device Innovation Consortium (MDIC).

BSI 2018. "Cyber Security Requirements for Network-Connected Medical Devices". https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/ICS/Medical_Devices_CS-E_132.html. German Federal Office for Information Security (BSI), Accessed Mar. 14, 2022.

Burns, A. J., M. E. Johnson, and P. Honeyman. 2016. "A Brief Chronology of Medical Device Security". *Communications of the ACM* vol. 59 (10), pp. 66–72.

Carreon, N. A., S. Lu, and R. Lysecky. 2021, jan. "Probabilistic Estimation of Threat Intrusion in Embedded Systems for Runtime Detection". *ACM Trans. Embed. Comput. Syst.* vol. 20 (2), pp. 1–27.

Chen, G. and Kotz, D. 2000. "A Survey of Context-Aware Mobile Computing Research". http://cs.dartmouth.edu/~dfk/papers/chen:survey-tr.pdf. Accessed Mar. 14, 2022.

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. Cybersecurity and Infrastructure Security Agency (CISA).

Claroty 2022. "Biannual ICS Risk & Vulnerability Report: 2H 2021". https://claroty.com/2h21-biannual-report/. Accessed Mar. 14, 2022.

Easttom, C., and N. Mei. 2019. "Mitigating Implanted Medical Device Cybersecurity Risks". In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pp. 0145–0148. Columbia University, New York City, USA.

FDA 2005. "Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices". https://www.fda.gov/media/73065/download. U.S. Food and Drug Administration (FDA). Accessed Mar. 14, 2022.

FDA 2016a. "Medical Devices – Classify Your Medical Device.". http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice. U.S. Food and Drug Administration (FDA). Accessed Mar. 14, 2022.

FDA 2016b. "Postmarket Management of Cybersecurity in Medical Devices". https://www.fda.gov/media/95862/download. U.S. Food and Drug Administration (FDA). Accessed Mar. 14, 2022.

FDA 2018. "Content of Premarket Submissions for Management of Cybersecurity in Medical Devices". https://www.fda.gov/media/119933/download. U.S. Food and Drug Administration (FDA). Accessed Mar. 14, 2022.

Gartner 2021. "Gartner Predicts By 2025 Cyber Attackers Will Have Weaponized Operational Technology Environments to Successfully Harm or Kill Humans". https://www.gartner.com/en/newsroom/press-releases/2021-07-21-gartner-predicts-by-2025-cyber-attackers-will-have-we. Accessed Mar. 17, 2022.

Lu, S., and R. Lysecky. 2019. "Data-driven Anomaly Detection with Timing Features for Embedded Systems". *ACM Transactions on Design Automation of Electronic Systems* vol. 24 (3), pp. 1–27.

Lu, S., M. Seo, and R. Lysecky. 2015. "Timing-Based Anomaly Detection in Embedded Systems". In *20th Asia and South Pacific Design Automation Conference*, pp. 809–814. Chiba/Tokyo, Japan.

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. Accessed Mar. 16, 2022.

Medtronic 2021. "Urgent Medical Device Communication Notification Letter Medtronic HVAD™ System". https://www.medtronic.com/content/dam/medtronic-com/global/HCP/Documents/hvad-urgent-medical-device-notice-june-2021.pdf. Accessed Mar. 14, 2022.

NRC 2019. "Standard Technical Specifications – Operating and New Reactors – Current Versions". https://www.nrc.gov/reactors/operating/licensing/techspecs/current-approved-sts.html. U.S. Nuclear Regulatory Commission (NRC). Accessed Mar. 11, 2022.

Palo Alto Networks 2022. "Know Your Infusion Pump Vulnerabilities and Secure Your Healthcare Organization". https://unit42.paloaltonetworks.com/infusion-pump-vulnerabilities/. Accessed Mar. 14, 2022.

Rao, A., J. Rozenblit, R. Lysecky, and J. Sametinger. 2018, Apr. "Trustworthy multi-modal framework for life-critical systems security". In *Proceedings of the Annual Simulation Symposium*, ANSS '18, pp. 1–9, Society for Computer Simulation International. San Diego, CA, USA.

Riegler, M., and J. Sametinger. 2020. "Mode Switching from a Security Perspective: First Findings of a Systematic Literature Review". In *Database and Expert Systems Applications*, pp. 63–73, Springer. Bratislava, Slovakia.

Riegler, M., J. Sametinger, M. Vierhauser, and M. Wimmer. 2022. "Automatic Mode Switching based on Security Vulnerability Scores". *submitted for publication*.

Ross, R., M. McEvilley, and J. Carrier Oren. 2016. *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*. National Institute of Standards and Technology (NIST).

Sametinger, J., and J. Rozenblit. 2016. "Security Scores for Medical Devices". In *9th International Conference on Health Informatics*, pp. 533–541. In Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016) - Volume 5: HEALTHINF. Porto, Portugal.

Sametinger, J., J. Rozenblit, R. Lysecky, and P. Ott. 2015. "Security Challenges for Medical Devices". *Communications of the ACM* vol. 58 (4), pp. 74–82.

Sametinger, J., and C. Steinwender. 2017. "Resilient Context-Aware Medical Device Security". In *International Conference on Computational Science and Computational Intelligence, Symposium on Health Informatics and Medical Systems (CSCI-ISHI)*, pp. 1775–1778. IEEE Computer Society. Las Vegas, NV, USA.

Samie, F., V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, and J. Henkel. 2019, Mar. "Oops: Optimizing Operation-mode Selection for IoT Edge Devices". *ACM Transactions on Internet Technology* vol. 19 (2), pp. 22:1–22:21.

SmartCockpit 2004. "A330-A340 Flight Crew Training Manual". https://www.smartcockpit.com/docs/A330-A340_Flight_Crew_Training_Manual_1.pdf. Accessed Mar. 14, 2022.

TCG 2019. "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. Trusted Computing Group (TCG). Accessed Mar. 11, 2022.

AUTHOR BIOGRAPHIES

MICHAEL RIEGLER is a PhD Student in the LIT Secure and Correct Systems Lab at the Johannes Kepler University Linz and is currently a Research Associate at the Department of Electrical and Computer Engineering at the University of Arizona. He works on medical and industrial security and design and worked for several years for an industrial company. He received his BSc and MSc in Business Informatics focusing on Security Engineering and Management from the same university in 2011 and 2014. His email address is michael.riegler@jku.at.

JERZY W. ROZENBLIT is a University Distinguished Professor, Raymond J. Oglethorpe Endowed Chair in the Electrical and Computer Engineering (ECE) Department, with a joint appointment as Professor of Surgery in the College of Medicine at the University of Arizona. During his tenure at the University of Arizona, he established the Model-Based Design Laboratory with major projects in design and analysis of complex, computer-based systems, hardware/software codesign, and simulation modeling. He presently serves as Director of the Life-Critical Computing Systems Initiative, a research enterprise intended to improve the reliability and safety of technology in healthcare and life-critical applications. His email address is jerzyr@arizona.edu.

JOHANNES SAMETINGER is an Associate Professor at the Department of Information Systems – Software Engineering and the LIT Secure and Correct Systems Lab at the Johannes Kepler University Linz, Austria. He holds a Dr. techn. in Computer Science from the same university. His research interests include software engineering and IT security with an emphasis on software security and medical device security. He has made several research visits to universities in the United States and Canada, including the University of Arizona. His email address is johannes.sametinger@jku.at.