Probabilistic Threat Detection for Risk Management in Cyber-physical Medical Systems

Article *in* IEEE Software · January 2017

DOI: 10.1109/MS.2017.4541031

4 authors, including:

Jerzy W Rozenblit
The University of Arizona
270 PUBLICATIONS **1,575** CITATIONS

SEE PROFILE

Probabilistic Threat Detection for Risk Management in Cyber-physical Medical Systems

Aakarsh Rao, Nadir Carreón, Roman Lysecky, and Jerzy Rozenblit,
University of Arizona

// Medical devices are complex cyber-physical systems exposed to numerous security risks and vulnerabilities. This article presents a dynamic risk management and automated threat mitigation approach based on a probabilistic threat estimation framework. A smart-connected-pacemaker case study illustrates the approach. //

THE INTERNET OF THINGS (IoT) represents the foundation of radical changes in cyber-physical systems.[1] There is rapid development and incorporation of Internet-connected devices in our lives, transforming several fields. This has been possible due to technical advancements of incorporating efficient computational resources, advanced sensors, and networking capabilities that allow communication of devices with the Internet as well as other devices.[2] Unsurprisingly, the IoT is strongly influencing advances in healthcare and medical-device development. Such devices are now part of the digital-health ecosystem. They facilitate continual patient monitoring and service, interoperability, and real-time data access. However, several critical challenges, including security, safety, privacy, essential performance, and regulatory compliance, have emerged.

Medical devices are exposed to a wide attack surface. Instances of malware, security vulnerabilities, and threats are proliferating. A significant number of recalls over the years have taken place.[3–5] In addition to strict regulations required for medical devices by the US Food and Drug Administration (FDA), recommendations for risk assessment and management for premarket and postmarket security management are now becoming standard.[6,7]

To ensure safety, security, and privacy in the presence of unknown security threats, devices should dynamically detect and assess risk, subsequently taking automated mitigative actions when the risk is elevated. This requires that a risk assessment model be developed at design time with runtime security threat detection, adaptive risk management policies, and automated mitigation schemes during deployment. Flexible security frameworks that incorporate conventional security solutions along with in-device security are required.[8] Toward this goal, we previously proposed a multimodal-device design with a composite risk model.[9] Here, we describe how we've incorporated a novel real-time threat detector with an adaptive risk assessment methodology to ensure unabridged threat mitigation during the deployment of devices.

 0740-7459/18/$33.00 © 2018 IEEE

Much work exists in real-time threat assessment and management, especially in intrusion detection systems.[10,11] Probabilistic methods like Markov models have been utilized to detect threats in such systems.[12] However, in critical medical cyber-physical systems that are characterized by strict timing constraints, expedient and robust threat detection is essential.[13] This necessitates the analysis of the distribution of events in each execution window compared to the current state sample as in Markov models. Thus, we utilize cumulative distribution functions (CDFs) for modeling the normal device behavior, which is used to quantify the likeliness of security threats at runtime. This probabilistic threat detector is used to assess and manage the system's risk, which results in a precise real-time update of the current system risk. This approach reduces the false-positive rate to prevent erroneous activation of a mitigation scheme that may otherwise lead to accidental loss of functionality. This article presents a comprehensive framework for threat detection and mitigation during deployment of medical devices. We demonstrate this framework through a smart-connected-pacemaker scenario.

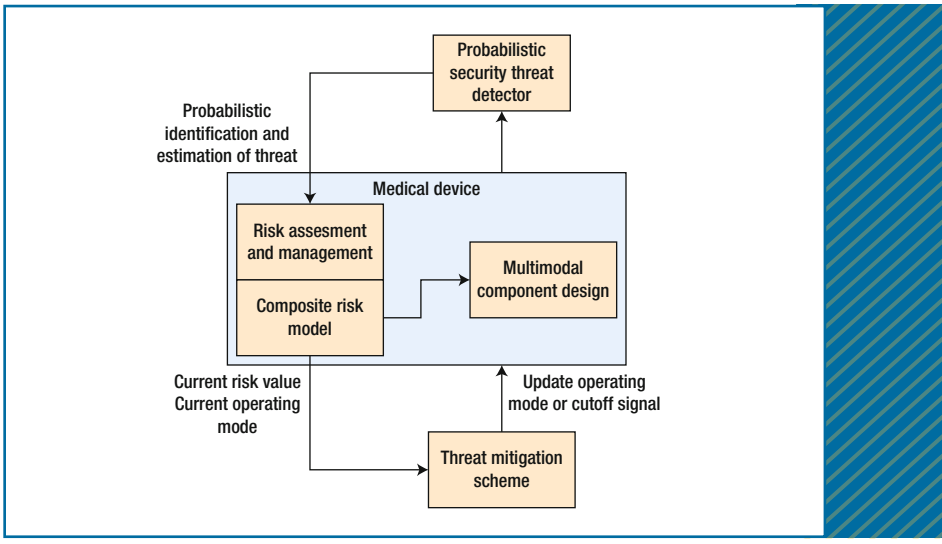

FIGURE 1. Overview of a framework for runtime threat detection, risk-based assessment, and automated mitigation in medical devices.

Framework Design Overview

An overview of our approach is presented in Figure 1. The medical device has been designed based on our multimodal approach. The composite risk model associates risk values with the device's various software and hardware components. For the details of the composite risk model and multimodal design, we direct readers to "Composite Risk Modeling for Automated Threat Mitigation in Medical Devices."[9] Based on the current system risk, which will be updated dynamically, the threat mitigation either disallows access to the affected component or updates the current operating mode to mitigate the risk while sustaining essential functionality. In this article, we focus on the integration of threat detection with risk assessment and management during medical-device deployment.

Threat Detection Design

The runtime threat detector monitors the execution sequences and timing of all critical system operations, specifically those within the system's composite risk model. The threat detection analyzes the timing of these system operations within a sliding execution window. For each execution window E_w, the CDF is calculated and compared to predefined bounds of the system's normal execution behavior. Using a CDF-based model of the system behavior under normal execution scenarios enables the runtime threat detection to estimate the presence of a threat affecting each operation. Utilizing the internal execution time provides protection against cloaked threats that follow the correct execution sequence but whose behaviors still have an impact on the operation execution time, a feature lacking in sequence-only detectors. This estimation compares the overlap between the CDFs obtained at runtime and the CDFs obtained from the system under normal circumstances.

To construct the normal execution model, the software application is statically analyzed to identify the critical operations defined in the device's composite risk model. The system is executed under different execution scenarios, and timing measurements are collected for all operations. The timing of the operations is obtained automatically and nonintrusively through the system's trace port (in our case study, the pacemaker's), which, importantly, does not perturb the device's execution. The CDF analysis checks the timing

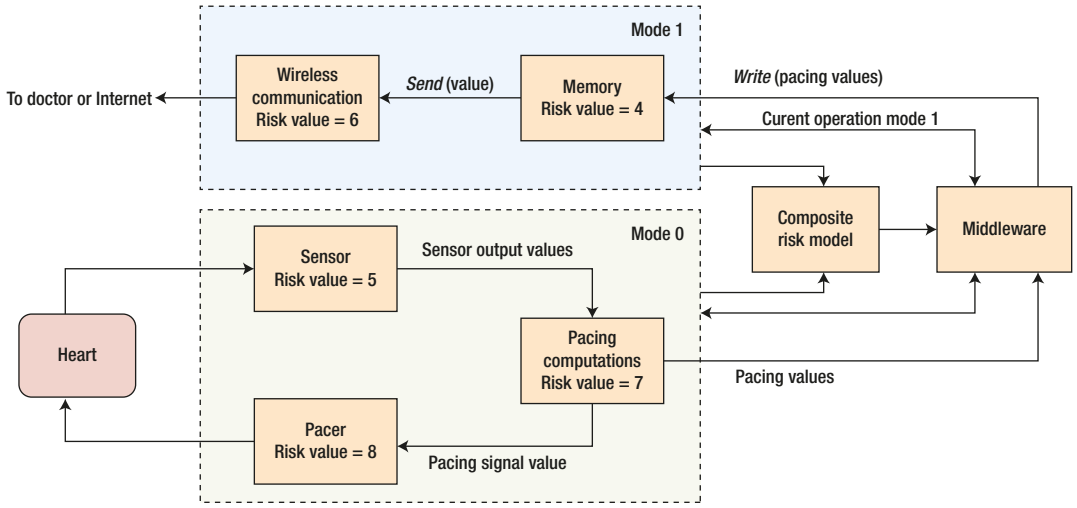

FIGURE 2. A multimodal smart connected pacemaker with its composite risk model.

across all execution windows for the training data and calculates the upper and lower distribution bounds for each operation. This is used at runtime to detect any deviation from the expected execution. For a single operation, the estimated threat probability depends on the complement of the overlap between the runtime CDFs and the CDF boundaries of the normal system execution. Finally, to eliminate or minimize false positives, cross-validation is used to determine the maximum estimated threat for normal operation execution. This threshold P_{th} is used to filter out false positives at runtime.

Risk Assessment and Management Unit

The estimated threat probabilities from the runtime threat detection are directly utilized as the input of our risk assessment to update the risk values of system components and operations. We use a level-based approach to update the risk for individual system operations based on the estimated threat probability. Risk values are updated as follows:

$$risk_{updated} = risk_{initial} + P_t \times C_l, \quad (1)$$

where P_t represents the estimated threat probability affecting the component and C_l is the level-based constant. During device design, every component is assigned an initial base risk value. This is determined by the criticality of the component or an expert's judgment, which is updated during deployment according to Equation 1. C_l is deduced as

$$C_l = \begin{cases} 0, & P_t < P_{th} \\ \left\lceil \frac{P_t}{P_{th}} \right\rceil, & otherwise \end{cases},$$

where P_{th} represents the probability thresholds. As P_t increases, $risk_{updated}$ will increment faster, and if it is slightly above the threshold (still a security threat), the risk will increase at a slower rate. The formulation also restricts the risk from increasing too rapidly by truncating C_l to the smallest succeeding integer value. If the threat persists, the cumulative risk will continue increasing, relative to the threat probability, until the device's operating threshold is reached. At this point, the mode of operation is switched to a lower mode.

A key aspect to consider while incorporating our proposed framework is to assure that the latency of the overall risk management is well within the temporal limits of activating the principal intended action by the medical device.[14] Thus, once the threat is detected, the mitigation latency L is calculated as

$$L = n \times t_{update} + t_{mode}, \quad (2)$$

where n is the number of windows that were analyzed between the time when the threat was introduced in the system and the time when the current-mode maximum risk was reached, t_{update} is the time

required for the risk to be updated, and t_{mode} is the time to switch modes. n depends on multiple factors, mainly the threat and execution window E_w. In the real world, there is no method to precisely know when the threat was introduced into the system. Therefore, the latency is calculated by conducting experiments and measuring the estimated probability of the threats. Our threat mitigation response adapts according to the measured estimated probability.

Smart-Connected-Pacemaker Scenario

We developed a smart-connected-pacemaker prototype and implanted malware therein to demonstrate our framework.[13] Figure 2 shows the pacemaker design, based on the multimodal approach with the composite risk model as described in "Composite Risk Modeling for Automated Threat Mitigation in Medical Devices."[9] For our demonstration, we consider two operational modes, but we note that our framework can accommodate any number of modes as required by the designer. We model higher levels of abstraction for our device components (e.g., Bluetooth or WiFi would both be included in the wireless-communication component) to emphasize the operation of our proposed framework.

The critical components required for the pacemaker's essential performance include the pacer, sensor, and pacing-computations component, which are incorporated in Mode 0. The other components are used in Mode 1, as they do not contribute to the essential functionality. Hardware–software middleware facilitates the secure transfer of data and signals between operational modes. The middleware is also responsible for analyzing the runtime threat detection, updating the risk model, and determining what mitigation strategy to invoke when a threat is detected. The device is assumed to run with full functionality in the highest mode. Note that the cumulative risk for Mode 0 is 20 and the cumulative risk for Mode 1 is 30.

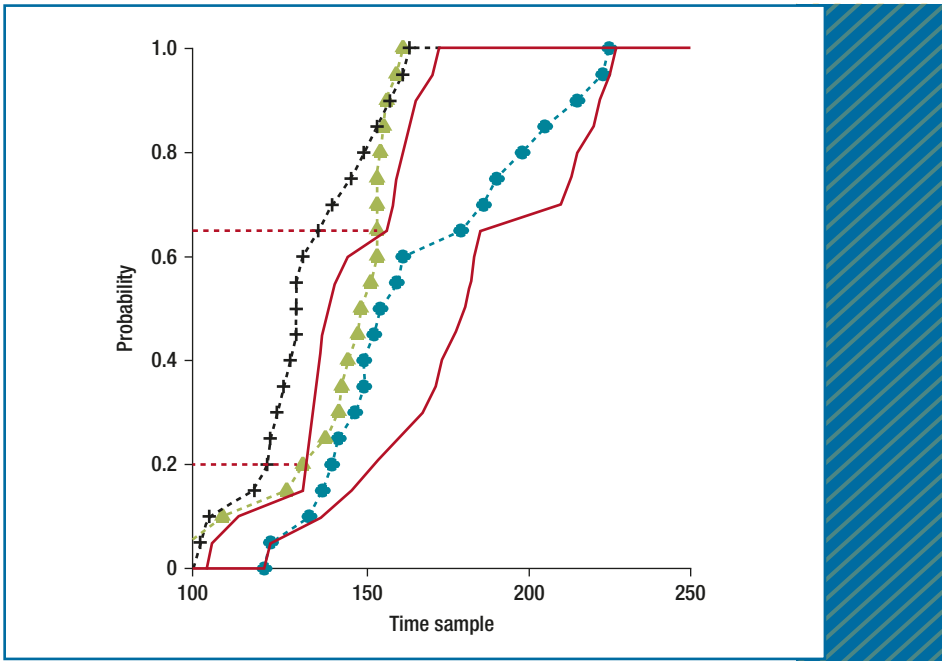

FIGURE 3. An example of CDF-based threat estimation based on real data from the smart-connected-pacemaker prototype. The red line represents CDF (cumulative distribution function) bounds. The black, blue, and green lines are runtime CDFs with estimated threat probabilities of 100%, 0%, and 55%, respectively.

Making the value of C_l dependent on both the estimated threat probability and the threshold allows the system to increase the risk at either a faster or slower rate for different scenarios. Figure 3 presents how the threat probability is calculated in our scenario. The red solid line represents the CDF bounds for the normal execution model. The black, blue, and green lines represent the CDFs for three runtime execution windows. The black CDF is completely outside the boundaries and thus has an estimated threat probability of 100%. In contrast, the blue CDF is completely inside the boundaries, and thus the threat estimate is 0%. For the green CDF, there is partial overlap with the predefined boundaries, and the probability is estimated as the percentage of points of the CDF that fall outside the boundaries. The threat probability is equal to $1 - (0.65 - 0.20)$, or 0.55, indicating there is an estimated 55% chance of a threat.

An illustrative example shows how the shift in modes is done based on the estimated threat probability. Using Figure 2 as a starting point for our example and Mode 1 as the initial operating mode, we can observe that the wireless-communication component has a risk value of 6 for the current execution. For simplicity,

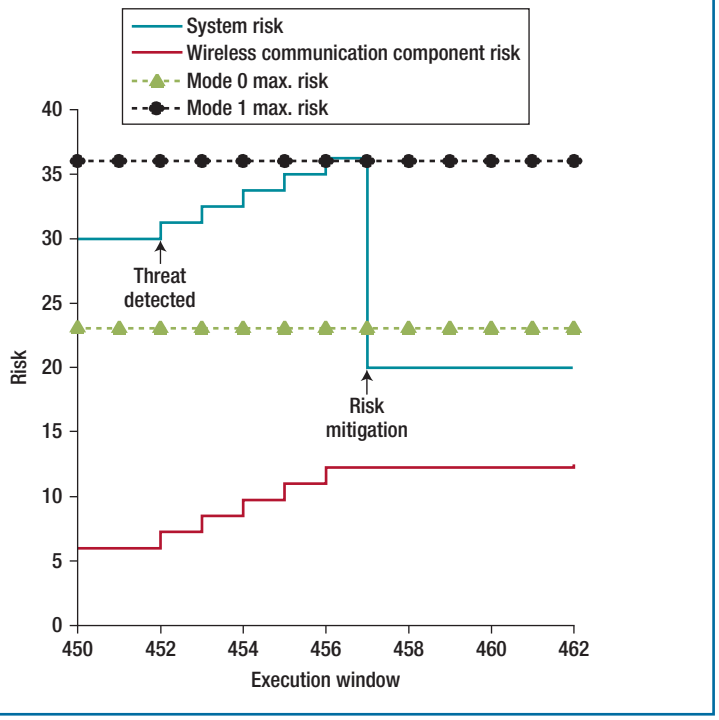

FIGURE 4. Mode switch scenario.

we consider malware that only affects the wireless-communication component where $P_{th}(wireless) = 5\%$.

From our conducted experiment, Figure 4 shows how the wireless-communication component's risk (the red line) increases over time, based on the estimated threat probability. As the threat persists, the risk continues to increase, until the cumulative risk (the blue line) exceeds Mode 1's maximum risk threshold (the black line). In response, the middleware mitigates the risk by transitioning to Mode 0, thereby reducing the overall system risk. Additionally, the affected component is no longer used in Mode 0.

In this scenario, runtime threat detection is performed on an execution window corresponding to five iterations of the communication component. As such, the detection latency of five execution windows is equivalent to 25 iterations of the communication thread. The principal intended action of a pacemaker is to trigger a pulse to ensure a normal heart rate from 60 to 100 beats per minute, translating to 1 beat every 1 to 0.6 seconds. Utilizing Equation 2, the total threat detection and mitigation latency L is approximately 375 ms, which is well within the lower threshold time of 600 ms to trigger a normal beat.

In conclusion, we have briefly described and exemplified our approach for efficient runtime security threat detection, dynamic risk assessment, and automated mitigation for medical devices.

References

1. K. Carruthers, "Internet of Things and Beyond: Cyber-physical Systems," *IEEE Internet of Things Newsletter*, 10 May 2014; iot.ieee.org/newsletter/may-2016/internet-of-things-and-beyond-cyber-physical-systems.html.
2. K. Rose, S. Eldridge, and L. Chapin, *The Internet of Things (IoT): An Overview*, Internet Soc., 2015; www.internetsociety.org/resources/doc/2015/iot-overview.
3. D.B. Kramer et al., "Security and Privacy Qualities of Medical Devices: An Analysis of FDA Postmarket Surveillance," *PLOS ONE*, vol. 7, no. 7, 2012; journals.plos.org/plosone/article?id=10.1371/journal.pone.0040200.
4. J. Radcliffe, "Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System," presentation at 2011 Black Hat Conf, 2011.
5. D. Halperin et al., "Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses," *Proc. 2008 IEEE Symp. Security and Privacy* (SP 08), 2008, pp. 129–142.
6. *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices*, US Food and Drug Administration, 2005; www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089593.pdf.
7. *Postmarket Management of Cybersecurity in Medical Devices*, US Food and Drug Administration, 2016; www.fda.gov/downloads/Training/CDRHLearn/UCM537944.pdf.
8. S. Babar et al., "Proposed Embedded Security Framework for Internet of Things (IoT)," *Proc. 2011 2nd Int'l Conf. Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic*

Systems Technology (Wireless VITAE 11), 2011, pp. 1–5.
9. A. Rao et al., "Composite Risk Modeling for Automated Threat Mitigation in Medical Devices," *Proc. 2017 Spring Simulation Multiconf.* (SpringSim 17), 2017.
10. A. Blyth and P. Thomas, "Performing Real-Time Threat Assessment of Security Incidents Using Data Fusion of IDS Logs," *J. Computer Security*, vol. 14, no. 6, 2006, pp. 513–534.
11. Y. Cherdantseva et al., "A Review of Cyber Security Risk Assessment Methods for SCADA Systems," *Computers and Security*, vol. 56, Feb. 2016, pp. 1–27.
12. A. Arnes et al., "Using Hidden Markov Models to Evaluate the Risks of Intrusions: System Architecture and Model Validation," *Recent Advances in Intrusion Detection*, LNCS 4219, 2006, pp. 145–164.
13. S. Lu, M. Seo, and R. Lysecky, "Timing-Based Anomaly Detection in Embedded Systems," *Proc. 20th Asia and South Pacific Design Automation Conf.*, 2015, pp. 809–814.
14. J.L. Martin et al., "Medical Device Development: The Challenge for Ergonomics," *Applied Ergonomics*, vol. 39, no. 3, 2008, pp. 271–283.

ABOUT THE AUTHORS

AAKARSH RAO is a PhD student in computer engineering at the University of Arizona. His research interests are secure-device design, medical-device security, hardware–software codesign, and path planning for surgical systems. Rao received an MS in electrical and computer engineering from the University of Arizona. Contact him at aakarshrao7@email.arizona.edu.

NADIR CARREÓN is a PhD student in electrical and computer engineering at the University of Arizona. His research interests include embedded systems, with an emphasis on security and threat detection in medical devices. Carreón received a bachelor of mechatronics engineering from Universidad de Sonora. Contact him at nadir@email.arizona.edu.

ROMAN LYSECKY is an associate professor of electrical and computer engineering at the University of Arizona. His research focuses on embedded systems, with an emphasis on medical-device security, automated threat detection and mitigation, runtime adaptable systems, performance and energy optimization, and nonintrusive observation methods. Lysecky received a PhD in computer science from the University of California, Riverside. Contact him at rlysecky@ece.arizona.edu.

JERZY ROZENBLIT is the University Distinguished Professor, Raymond J. Oglethorpe Endowed Chair, in the University of Arizona's Electrical and Computer Engineering Department, and he holds a joint appointment as a professor of surgery in the university's College of Medicine. Jointly with the Arizona Simulation Technology and Education Center, he's developing computer-guided training methods and systems for minimally invasive surgery. Rozenblit received a PhD in computer science from Wayne State University. He's the director of the Life-Critical Computing Systems Initiative and a Fellow of the Society for Modeling and Simulation. Contact him at jr@ece.arizona.edu.