Tracing Techniques for Connected Medical Devices

Vaios Bolgouras
University of Piraeus
Piraeus, Greece

Georgios Petychakis
University of Piraeus
Piraeus, Greece

Aristeidis Farao
University of Piraeus
Piraeus, Greece

Apostolis Zarras
University of Piraeus
Piraeus, Greece

Christos Xenakis
University of Piraeus
Piraeus, Greece

Abstract

Integrating Connected Medical Devices (CMDs) into modern healthcare systems has enhanced clinical workflows and improved patient care through real-time data sharing. However, this digital evolution introduces significant security and privacy risks. Ensuring the integrity and trustworthiness of these devices is critical, as vulnerabilities may compromise patient safety and the healthcare infrastructure. This chapter investigates the role of tracing technologies in safeguarding CMDs, specifically in monitoring security and performance across both high-end and low-end devices. Advanced tracing tools like the extended Berkeley Packet Filter (eBPF) enable continuous monitoring and anomaly detection in high-end CMDs. In contrast, resource-constrained low-end CMDs necessitate a balance between security and performance through tailored tracing solutions. We explore static and dynamic properties vital for maintaining device integrity, such as secure boot processes and real-time operational data, and discuss the unique challenges of securing low-end devices with limited resources. Beyond device-level monitoring, tracing technologies contribute to the broader security lifecycle, enabling early detection of vulnerabilities, regulatory compliance, and forensic analysis in case of a breach. The chapter concludes with insights into future trends in CMD tracing technologies, addressing emerging threats while maintaining a balance between security, performance, and resource efficiency.

1 Introduction

Integrating *Connected Medical Devices* (CMDs) into modern healthcare systems has fundamentally changed the patient care, diagnosis, and treatment landscape. These devices, from wearable health monitors to complex implantable technologies, enable real-time monitoring, data collection, and communication with healthcare providers, supporting more accurate clinical decisions and personalized care. As the healthcare industry embraces the digital revolution, CMDs play a pivotal role in bridging the gap between patient needs and medical expertise, often functioning autonomously or as part of larger medical systems [1]. However, the growing reliance on these devices is not without challenges, particularly in security and privacy.

The increased connectivity and functionality of CMDs introduce new vulnerabilities, making them an attractive target for cyberattacks. These devices often handle sensitive patient data, including personal health information, which must be protected to ensure patient privacy [2]. Furthermore, the integrity and availability of CMDs are critical; any compromise in their functionality can have dire consequences, especially when a device is responsible for life-sustaining tasks such as monitoring vital signs or delivering medication. Thus, ensuring the trustworthiness of CMDs is paramount, and the ability to detect, respond to, and mitigate potential threats is a central concern for healthcare providers, regulatory bodies, and device manufacturers alike.

The challenge of securing CMDs is compounded by the diversity of devices in use today. High-end devices, such as those used in hospitals for intensive monitoring, have significant computational resources and advanced capabilities, making them well-suited to sophisticated security solutions. These devices can support robust security protocols and complex tracing mechanisms, enabling continuous system behavior monitoring [3]. In contrast, low-end devices, such as wearable monitors or home-use medical equipment, are often constrained by limited processing power, memory, and battery life. These resource limitations make it difficult to implement traditional security solutions, necessitating the development of tailored approaches that balance security needs with operational efficiency [4].

Tracing technologies have emerged as a critical solution in this context, continuously monitoring CMDs' behavior. Tracing refers to observing a device's operations in real time, recording static properties (e.g., secure boot processes) and dynamic behaviors (e.g., system calls, network activity, and performance metrics). By tracking this data, tracing technologies enable early detection of anomalies or deviations from expected behavior, providing valuable insights into the device's operational integrity and security posture. These insights are essential for maintaining the trustworthiness of CMDs, allowing for proactive responses to potential threats before they can impact patient care.

For high-end CMDs, tracing is often implemented using sophisticated technologies like the extended eBPF, which allows for deep system monitoring at the kernel level. eBPF provides a flexible and efficient means of tracking a wide range of system activities, including network traffic, filesystem changes, and process executions, without introducing significant performance overhead [5]. This capability makes it particularly well-suited for high-end medical devices requiring robust security and high performance.

On the other hand, low-end devices present unique challenges regarding tracing. These devices often operate on RTOS, designed to manage time-sensitive tasks with minimal delay [6]. Tracing on low-end devices must account for the limited computational resources available, ensuring that monitoring activities do not interfere with the device's primary functions. Static tracing methods, which monitor predefined behaviors such as boot integrity, and dynamic tracing methods, which observe real-time performance, must be carefully balanced to maintain security and operational efficiency.

This chapter delves into the critical role of tracing technologies in ensuring the security and integrity of CMDs across both high-end and low-end devices. Towards this direction, the ENTRUST framework is proposed, specifically designed to establish trust in CMDs through a robust architecture that integrates real-time monitoring, attestation, and verification mechanisms. It provides a comprehensive approach to managing the security of CMDs by leveraging tracing technologies to assess the behavior and operational integrity of these devices continuously. A key focus of ENTRUST is to address the specific challenges and solutions associated with tracing in different device categories, providing a comprehensive overview of the techniques and technologies that support secure device operation. By exploring the technological underpinnings of ENTRUST's tracing approach, the chapter sheds light on the importance of continuous monitoring for maintaining the trustworthiness of CMDs in an increasingly connected healthcare ecosystem. Additionally, this chapter addresses how tracing fits into the broader context of CMD security, operational assurance, and compliance with regulatory standards, ensuring that healthcare systems can continue to rely on these devices without compromising patient safety.

2 Tracing

Tracing is indispensable in the ENTRUST framework, serving as a core component for monitoring and ensuring the security, privacy, and integrity of CMDs. As CMDs become more critical to healthcare delivery and patient management, their secure operation must be continuously assessed to detect potential threats, identify vulnerabilities, and ensure compliance with strict regulatory standards. The complexity of this task is heightened by the diversity of CMDs, which range from powerful hospital-grade devices to more resource-constrained wearable or home-use devices. ENTRUST addresses this challenge by leveraging tailored tracing technologies that meet the specific needs of both high-end and low-end devices.

In ENTRUST, tracing is defined as continuously observing and recording a device's behavior, including its static properties (e.g., secure boot sequences) and dynamic operations (e.g., runtime performance and network activity). The goal is to establish a comprehensive, real-time understanding of the device's operational state, which can then be used to verify the integrity of the device, ensure compliance with security policies, and detect any abnormal or unauthorized behaviors that could indicate a security breach. Tracing is a defensive measure and a proactive tool, enabling early detection of vulnerabilities and potential exploits before they can compromise the device or the broader healthcare network. For CMDs, tracing typically involves monitoring a wide range of system components, including hardware, software, network interfaces, and application layers. By tracing these components, ENTRUST can gather critical data as the foundation for trust assessments. These assessments evaluate the device's overall security posture, determining whether it meets the required trust level for operation within a healthcare network. If a device's behavior deviates from expected norms or an anomaly is detected, ENTRUST's tracing mechanisms provide the necessary evidence to trigger corrective actions, such as isolating the device, applying software patches, or updating security configurations.

One of the most significant advantages of tracing technologies within ENTRUST is their ability to adapt to different CMDs, offering tailored solutions for high-end and low-end devices. High-end CMDs, such as hospital monitoring systems and complex diagnostic equipment, benefit from sophisticated tracing technologies like the eBPF. These technologies enable deep inspection of system operations, allowing for comprehensive monitoring of kernel-level activities, network communications, filesystem access, and process execution. High-end devices often have the computational resources to support these advanced tracing capabilities, making it possible to maintain continuous and detailed monitoring without degrading the device's performance. Conversely, low-end devices, typically used in less resource-intensive settings, require more efficient and lightweight tracing solutions. These devices, such as wearable heart monitors or glucose sensors, often operate with limited processing power, memory, and energy resources, necessitating a careful balance between security monitoring and performance efficiency. In such cases, tracing focuses on monitoring key system activities without overloading the device's limited resources. Techniques such as static tracing, where predefined behaviors are monitored, or dynamic tracing, where real-time performance is observed, are carefully applied to ensure that security is maintained without impacting the device's primary functions.

ENTRUST's tracing solutions are not limited to monitoring the internal operations of CMDs but also extend to interactions with external systems. CMDs frequently communicate with other devices, cloud platforms, and healthcare networks to share data, receive updates, or relay real-time information to clinicians. These interactions can introduce additional security risks, especially if the communication channels are not adequately protected. To address this, ENTRUST's tracing technologies monitor network traffic and data exchanges, ensuring that all communications are secure, encrypted, and in line with the device's trust policies. This holistic approach to tracing not only safeguards the device itself but also the broader network in which it operates. Additionally, the integration of tracing technologies within the ENTRUST framework

supports forensic analysis in the event of a security incident. By maintaining detailed logs of system operations and interactions, ENTRUST enables investigators to trace back through the device's activities, identifying the root cause of an attack or failure. This forensic capability is essential for understanding how and why a device was compromised, allowing for more targeted remediation efforts and improving the overall security of the CMD ecosystem.

2.1 Tracing in High-End Devices

High-end CMDs are typically found in hospitals and include complex diagnostic tools, advanced imaging systems, and real-time monitoring equipment. These devices are integral to critical care and rely on robust computational resources to perform complex functions such as processing large amounts of data, maintaining real-time patient monitoring, and integrating with broader healthcare information systems. Given their critical nature and data sensitivity, high-end CMDs demand highly effective and continuous monitoring mechanisms to ensure they remain secure and trustworthy throughout their operational lifecycle.

eBPF is a powerful technology adopted in the ENTRUST framework to provide sophisticated tracing capabilities for high-end devices. Originally designed for packet filtering in network operations, eBPF has evolved into a versatile tool that allows for deep inspection of a system's behavior by monitoring events at the kernel level. eBPF operates within the operating system kernel, making it possible to observe a wide range of system activities without requiring modification of the core application code or causing significant performance overhead. One of the key advantages of eBPF in high-end devices is its ability to execute user-defined programs in a secure, isolated environment within the kernel. This means that developers and system administrators can create custom monitoring functions tailored to the specific needs of a device. These programs can track events such as system calls, process execution, network activity, and filesystem interactions in real time. By tracing these events, eBPF helps ensure that the device operates as intended and that no unauthorized or anomalous behavior occurs.

In ENTRUST, eBPF supports real-time security monitoring and trust assessments for high-end CMDs. Its ability to dynamically track system behavior at the kernel level provides deep visibility into the device's operational state, allowing for the detection of potential security breaches or system malfunctions before they escalate into critical failures. The versatility of eBPF makes it especially well-suited for devices that require continuous monitoring and rapid response to emerging threats, as is often the case with life-critical medical equipment. Below we present some of the key benefits of eBPF for high-end devices:

Deep System Visibility. eBPF allows for detailed monitoring of kernel-level events, offering visibility into the operating system's inner workings. This is particularly useful for identifying low-level security issues, such as privilege escalation attacks, unauthorized system calls, or attempts to modify critical system files. eBPF can also be used to monitor performance metrics, such as CPU and memory usage, which ensure that high-end devices operate efficiently without being affected by potential malware or performance degradation.

Low Overhead Monitoring. Unlike traditional monitoring tools, eBPF operates with minimal overhead, ensuring that the performance of high-end devices is not significantly impacted by the tracing processes. This is crucial in the medical context, where high-end CMDs must perform complex tasks in real time, such as monitoring patients' vital signs or processing medical images, without experiencing slowdowns or interruptions due to security monitoring processes. The efficiency of eBPF results from its ability to run directly within the kernel, avoiding the need for frequent context switches between the kernel and user space. This reduces the resource consumption associated with tracing activities, making monitoring a wide range of system events possible without overwhelming the device's computational resources.

Programmability and Flexibility. One of eBPF's most significant advantages is its programmability. System administrators and developers can write custom eBPF programs to track specific events or behaviors relevant to a particular device or use case. This flexibility allows eBPF to be adapted to the unique requirements of high-end CMDs, whether they are monitoring patient data in real time or processing sensitive medical information. The flexibility of eBPF programs allows for creating highly targeted tracing mechanisms that focus on critical areas of the system, such as monitoring the integrity of executable code, network packet flows, or interactions between different software components.

Security and Isolation. eBPF runs in a highly secure and isolated environment within the kernel, ensuring that the tracing programs cannot interfere with the device's normal operation. This isolation is critical in medical devices, where any disruption in the system's functionality could have serious consequences for patient care. Additionally, eBPF's execution model ensures that user-defined programs are strictly controlled regarding what they can access and modify within the system, reducing the risk of introducing new vulnerabilities through tracing activities.

Real-Time Threat Detection. eBPF provides real-time insights into system behavior, enabling immediate detection of security anomalies such as unexpected process creation, unauthorized access attempts, or suspicious network activity. This real-time detection is essential for high-end CMDs, which must respond rapidly to any potential threats to maintain their operational integrity and ensure the safety of their patients. By continuously monitoring the device's activity, eBPF can trigger alerts or automated responses, such as isolating the device from the network, rolling back to a safe configuration, or initiating a security audit to identify and mitigate the threat.

Integration with Broader Security Tools. eBPF is often integrated with other security tools and frameworks to provide a comprehensive security posture for high-end CMDs. For example, data collected through eBPF tracing can be fed into machine learning algorithms or *Artificial Intelligence* (AI)-based anomaly detection systems, which analyze the data for patterns of malicious activity or performance issues. This integration allows for more proactive threat detection and response.

Figure 1 represents the architecture for integrating tracing in high-end within the ENTRUST framework. The architecture is structured into various layers corresponding to specific components and functionalities of the high-end device. It is organized based on trust boundaries, defined as Hardware, Trusted, and Untrusted regions. This delineation ensures secure operations and protection against potential threats. At the base of the architecture lies the hardware layer, which consists of the high-end device's physical components and includes a *Physical Unclonable Function* (PUF). The PUF is a unique hardware identifier, providing a strong foundation for device attestation and preventing hardware-level tampering. The secure bootloader is positioned above the hardware layer, which verifies the integrity and authenticity of the device's software components before allowing the system to boot. This ensures that only trusted software components are loaded during startup, establishing a secure foundation for subsequent operations. The *Operating System* (OS) layer forms the core of the software environment and is depicted with the eBPF technology integrated within it. eBPF is a flexible and low-overhead tracing mechanism that monitors kernel-level and application-level activities. This capability is critical for maintaining system integrity and detecting anomalous behaviors that could indicate security breaches or misconfigurations. Above the OS layer is the untrusted application layer, which includes two key components: the *Application* (APP) and the Trust Agent. The Trust Agent acts as an intermediary, communicating directly with the eBPF-based Tracer to send and receive security-related information, ensuring that the application behaves as expected. This layer is considered untrusted until the Tracer verifies the application's behavior as part of the system's

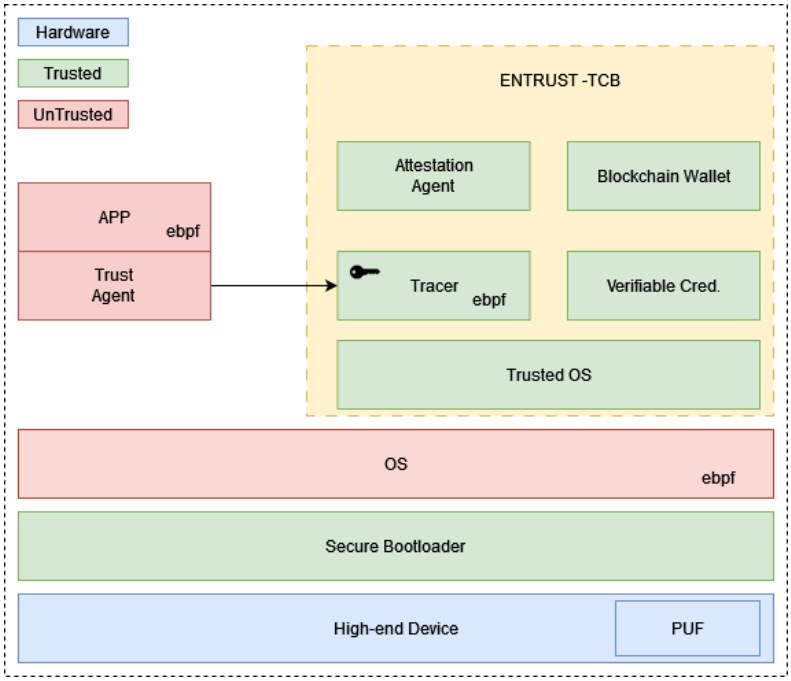

Figure 1: Architectural Overview of the ENTRUST Tracing Mechanism for High-End Devices

continuous monitoring and trust evaluation process.

The right section of Figure 1 showcases the *Trusted Computing Base* (TCB) within the ENTRUST framework. The TCB encompasses trusted software components running within a secure environment. These include the Attestation Agent, responsible for validating the security posture and integrity of the device, and the eBPF-based Tracer, which continuously monitors the CMD's behavior. The Tracer interacts with the Trust Agent in the untrusted layer, gathering data on system operations to provide a comprehensive view of the device's state. Additionally, the Blockchain Wallet component securely manages cryptographic credentials and supports secure transactions, while the Verifiable Credentials module stores cryptographically signed credentials that verify the CMD's identity and trustworthiness. Together, these components ensure that all activities within the TCB are securely managed and isolated from potentially compromised elements in the untrusted sections of the device. The Trusted OS, forming the base of the TCB, provides a secure execution environment for the attestation and verification processes. It ensures that sensitive operations are isolated from potentially untrusted or compromised components in the application layer and the standard OS environment, preserving the integrity of the device's security functions.

2.2 Tracing in Low-end Devices

While high-end CMDs benefit from robust computational resources supporting advanced tracing technologies like eBPF, low-end CMDs present different challenges. Often used in wearables, home monitoring systems, and other resource-constrained environments, these devices are essential for personalized care and remote health monitoring. However, they operate under stringent processing power, memory, and energy consumption limitations. This necessitates a more efficient and lightweight approach to tracing that ensures security without compromising the device's primary functionality. Low-end CMDs often run on *Real-Time Operating Systems* (RTOS) designed to manage time-critical tasks with minimal delays. RTOS is ideal for CMDs prioritizing responsiveness, such as monitoring vital signs or controlling drug delivery systems. However, the limited resources of these devices make it impractical to implement resource-intensive tracing

technologies like those used in high-end devices. Therefore, tailored solutions are required to balance security and performance while addressing the unique needs of low-end CMDs.

In low-end CMDs, RTOS plays a central role in managing device operations, ensuring that tasks are executed with precise timing to meet the stringent requirements of healthcare applications. For instance, a wearable heart monitor that tracks a patient's vital signs in real time must respond immediately to changes in heart rate or oxygen levels without delay. In such cases, any additional processes, such as tracing or security monitoring, must not interfere with the device's real-time performance. RTOS is designed to prioritize tasks, ensuring that critical functions, like monitoring and reporting vital signs, are given higher priority over less time-sensitive operations. This prioritization is crucial when incorporating tracing technologies, as security monitoring must coexist with the device's core functionality without causing performance bottlenecks.

The primary challenge of tracing in low-end CMDs is maintaining an optimal balance between security and resource efficiency. Given that these devices have limited processing power and memory, the overhead introduced by tracing activities must be minimal to avoid slowing down or disrupting the device's core functions. Furthermore, many low-end CMDs are battery-powered, meaning that energy consumption is a critical consideration. Any additional computational tasks, such as continuous tracing, could drain the device's battery faster, reducing its operational lifespan and requiring more frequent recharging or maintenance.

Another significant challenge is the reduced flexibility in implementing complex tracing mechanisms due to the hardware constraints of low-end devices. These devices often lack the advanced processing capabilities required for deep inspection of system behaviors, making it necessary to adopt more lightweight tracing solutions that focus on essential aspects of the system's operation.

Static and dynamic tracing techniques are commonly used to address the limitations of low-end CMDs, each offering distinct advantages depending on the specific use case.

Static Tracing. Static tracing involves monitoring predefined, unchanging aspects of the system, such as the integrity of the boot process, configuration settings, and firmware updates. These are typically fixed properties that do not change frequently and can be monitored efficiently without constant real-time inspection. For example, static tracing can ensure that the device's boot sequence follows a secure process, verifying the integrity of the firmware before allowing the device to operate. One of the key benefits of static tracing is its low overhead. Since it only tracks predefined, infrequent events or system states, static tracing consumes fewer resources than dynamic tracing, making it well-suited for low-end CMDs with limited processing power and memory. However, static tracing is less effective for detecting real-time security issues, such as runtime anomalies or unauthorized network communications, which may occur during the device's operation.

Dynamic Tracing. In contrast to static tracing, dynamic tracing focuses on monitoring the real-time behavior of a device during its operation. This type of tracing is particularly useful for detecting anomalies, such as unexpected system calls, abnormal network activity, or deviations from expected performance patterns. Dynamic tracing provides a more comprehensive view of the system's operational state, allowing for the detection of potential security threats as they occur. However, dynamic tracing introduces higher overhead than static tracing, as it requires continuous real-time monitoring of the system's behavior. For low-end CMDs, this presents a challenge, as their limited resources may not be able to support such intensive monitoring without impacting the device's performance. To mitigate this, dynamic tracing in low-end devices is often applied selectively, focusing on key system activities most critical to the device's security, such as monitoring network traffic or task execution.

One of the most critical aspects of implementing tracing in low-end CMDs is finding the right

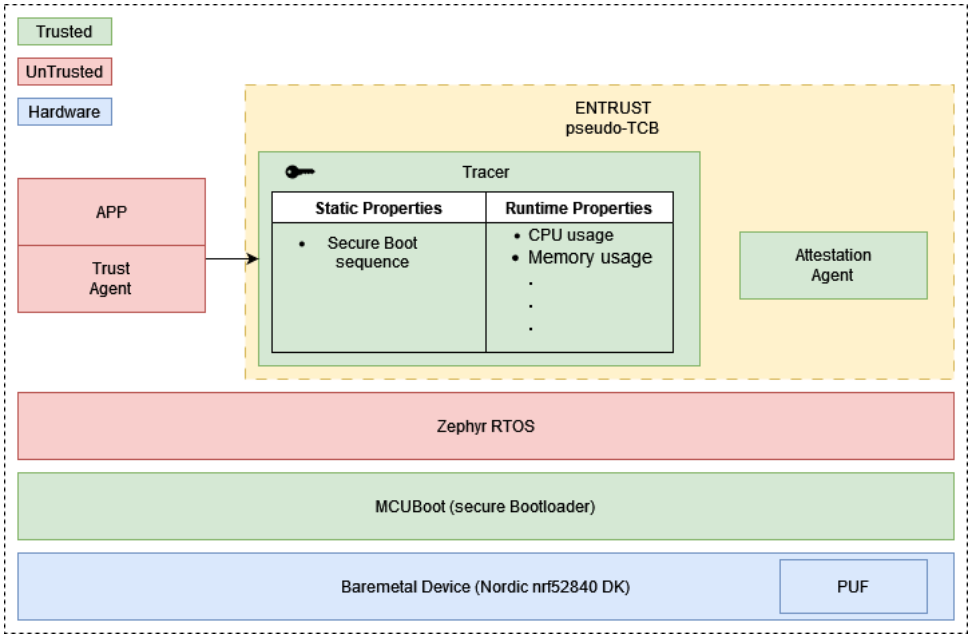

Figure 2: Architectural Overview of the ENTRUST Tracing Mechanism for Low-End Devices

balance between performance and security. In resource-constrained environments, monitoring every aspect of the system in real time is not feasible, as this would quickly overwhelm the device's processing capabilities and reduce its operational efficiency. Instead, a hybrid approach is often employed, combining static and dynamic tracing elements. For example, a low-end CMD may use static tracing to secure its boot process and firmware. In contrast, dynamic tracing is applied intermittently to monitor network activity or detect unusual behaviors during the device's operation. This approach allows continuous security monitoring without excessive demands on the device's limited resources. Additionally, event-driven tracing can further reduce the overhead associated with dynamic tracing. This approach triggers dynamic tracing only when specific events or conditions are met, such as when the device detects a sudden spike in network traffic or an unexpected system call. This ensures that tracing activities are focused on periods of high risk, minimizing the impact on the device's performance during normal operation.

To meet the unique needs of low-end CMDs, the ENTRUST framework incorporates lightweight tracing solutions designed specifically for resource-constrained environments. One such solution is the Segger SystemView, a real-time tracing tool commonly used in RTOS embedded systems and devices. SystemView allows for the efficient monitoring of real-time processes, task execution, and interrupt handling without introducing significant overhead. By leveraging tools like SystemView, ENTRUST can provide real-time visibility into the operation of low-end CMDs, ensuring that critical security events are detected and addressed promptly. These lightweight tracing solutions are designed to operate within the limitations of low-end devices, offering a practical balance between security and performance.

Figure 2 depicts the ENTRUST architecture designed for low-end devices structured to function in resource-constrained environments. The device hardware is represented by a Nordic nrf52840 *Development Kit* (DK) and a PUF, providing a unique hardware identifier to support secure device attestation and identity verification. The secure boot process is managed by the MCUBoot Secure Bootloader, which validates the integrity and authenticity of the firmware before allowing the device to initialize. Above the secure bootloader, the architecture includes the Zephyr RTOS, a lightweight OS commonly used in low-power embedded systems. The Zephyr RTOS manages the device's core operations, providing a foundation for real-time task scheduling, efficient re-

source management, and secure communication protocols. Zephyr allows the device to perform critical monitoring and control functions with minimal overhead, which is essential given the limited computational resources in low-end CMDs. The central component of this architecture is the ENTRUST Tracer, which is housed within a *pseudo-Trusted Computing Base* (pseudo-TCB). The Tracer monitors two categories of properties: (*i*) Static Properties and (*ii*) Runtime Properties. Static Properties include elements that do not change during operation, such as the secure boot sequence, ensuring that the device's initial state is verified and trusted. Runtime Properties cover dynamic aspects like CPU usage, memory consumption, and other real-time metrics that reflect the device's current state and performance. This dual monitoring approach allows the Tracer to detect anomalies at startup and during operation, providing comprehensive security coverage. To the left, Figure 2 shows the Trust Agent and the APP components. The Trust Agent interacts directly with the Tracer to communicate relevant security information, facilitating continuous monitoring and evaluation of the application's behavior. This ensures the device adheres to its security policies throughout its operational lifecycle. The architecture also includes the Attestation Agent within the pseudo-TCB, which validates the integrity and security posture of the device based on the evidence collected by the Tracer. The Attestation Agent's role is to generate verifiable claims regarding the device's trustworthiness, which can then be used to communicate its security status to external entities.

In healthcare, low-end CMDs are commonly used in applications such as remote patient monitoring, wearable health devices, and home-use diagnostic tools. These devices play a critical role in providing continuous care for patients outside of traditional clinical settings, often collecting vital health data that is transmitted to healthcare providers for analysis.

Wearable Heart Monitors. These devices continuously track a patient's heart rate and can alert healthcare providers if an irregular heartbeat is detected. Static tracing ensures the device's firmware is secure, while dynamic tracing monitors real-time data transmission to ensure that sensitive health information is not intercepted or altered.

Glucose Monitors. For patients with diabetes, wearable glucose monitors provide real-time feedback on blood sugar levels. Tracing in these devices focuses on maintaining the integrity of data transmissions, ensuring that readings are accurate and securely communicated to the patient's healthcare provider.

Home-Use Diagnostic Tools. Low-end CMDs used in home diagnostics, such as blood pressure monitors or pulse oximeters, require efficient tracing to ensure that their measurements are reliable and that abnormal readings are transmitted securely.

Tracing low-end CMDs presents unique challenges due to their limited resources and real-time operational requirements. By employing a combination of static and dynamic tracing, along with lightweight tools such as Segger SystemView, ENTRUST ensures that these devices can be monitored effectively without compromising performance. This approach allows low-end CMDs to remain secure and trustworthy, even in resource-constrained environments.

3 Impact

The use of tracing technologies in CMDs within the ENTRUST framework profoundly impacts the security, functionality, and trustworthiness of these devices. Tracing plays an essential role in safeguarding patient safety, ensuring regulatory compliance, and maintaining the overall integrity of the healthcare system. As CMDs become more deeply integrated into healthcare delivery, the need for robust monitoring solutions like tracing has grown, making these technologies crucial for operational assurance and long-term trust.

One of the most immediate impacts of tracing technologies is enhancing security across CMDs,

particularly in protecting patient data and maintaining device functionality. Tracing provides a powerful tool for real-time threat detection, allowing healthcare organizations and device manufacturers to continuously monitor CMDs for signs of security breaches, unauthorized access, or other abnormal behavior. In high-end devices, tools like the eBPF allow for deep system monitoring at the kernel level, providing immediate insight into system events such as unauthorized system calls, unexpected process execution, or malicious network traffic. The real-time nature of eBPF's monitoring capabilities ensures that potential security threats can be identified and addressed before they escalate into serious incidents, helping to prevent data breaches or system failures that could have catastrophic consequences for patient care.

For low-end devices with more limited resources, tracing solutions like static and dynamic tracing offer a more focused approach to security monitoring. By concentrating on key events such as secure boot verification or abnormal task execution, tracing ensures that these devices remain secure even in resource-constrained environments. The ability to detect anomalies early in low-end CMDs is particularly critical, as these devices often serve in less supervised environments, such as patient homes or remote monitoring setups, where direct oversight by healthcare professionals is limited.

The operational assurance of CMDs is another key area where tracing technologies have a significant impact. These devices often operate in real-time, managing or monitoring critical patient functions where reliability and uptime are essential. Tracing helps ensure that CMDs function properly by continuously monitoring system performance, detecting performance degradation, and verifying the integrity of key system components. In high-end CMDs, such as imaging equipment, surgical robots, or intensive care unit monitoring systems, performance monitoring through tracing can prevent system bottlenecks or failures that could interrupt critical medical services. Tracing enables healthcare providers to proactively maintain these systems, ensuring that performance remains within acceptable thresholds and that any emerging issues are addressed before they affect patient care. Low-end CMDs, such as wearable monitors or at-home diagnostic tools, also benefit from tracing by ensuring they function as intended over extended periods. For instance, dynamic tracing allows these devices to detect unusual system behavior or communication errors that might indicate a fault. This ensures that patients relying on these devices for continuous monitoring receive accurate and reliable readings, preventing delays in care or misdiagnoses due to faulty data collection or transmission.

In the healthcare sector, ensuring compliance with regulatory standards is critical for device manufacturers and healthcare providers. CMDs are subject to strict regulations regarding patient data handling, device operations' integrity, and network communications security. Tracing technologies are pivotal in helping organizations meet these regulatory requirements by providing continuous monitoring, generating audit logs, and offering real-time evidence of compliance. For example, tracing can ensure that CMDs comply with data protection regulations such as the *General Data Protection Regulation* (GDPR) by monitoring and securing the transmission of sensitive patient data. By continuously observing how a device handles data, tracing can provide assurance that privacy policies are being followed and that data is encrypted, anonymized, or securely transmitted as required by law. Furthermore, post-market surveillance regulations often require manufacturers to demonstrate that their devices operate securely and reliably after deployment. Tracing provides the necessary data for manufacturers to perform regular security assessments and issue software updates or patches to address new vulnerabilities. This ensures that CMDs comply with evolving regulatory requirements throughout their operational lifecycle.

In the event of a security breach or device malfunction, the ability to perform forensic analysis is critical for identifying the root cause of the problem and determining how to prevent future incidents. Tracing technologies enable comprehensive forensic analysis by capturing detailed

logs of system activities, network communications, and interactions between CMDs and external systems. These logs serve as a vital resource for investigating security incidents or device failures, allowing for reconstructing events that led to the issue. For high-end devices, eBPF's kernel-level tracing capabilities provide detailed insights into system behavior, making it easier to pinpoint the exact moment and cause of a security breach or system failure. This level of visibility is invaluable for incident response teams, as it allows them to trace the problem back to its origin, whether it was a malicious attack, a software bug, or a hardware failure. By providing a clear timeline of events, tracing helps to reduce the time required to diagnose and resolve issues, minimizing the impact on patient care. In low-end devices, where resources are more limited, event-driven tracing can still provide critical forensic data in the event of an incident. By recording key events such as boot sequences, task execution, or network communications, tracing ensures that investigators have access to the information they need to perform a thorough post-incident analysis. This capability is particularly important for CMDs that operate in remote or unsupervised environments, as it allows security teams to analyze incidents after the fact and implement preventive measures to protect against future occurrences.

Ultimately, the impact of tracing technologies on healthcare delivery and patient safety cannot be overstated. By ensuring that CMDs remain secure, reliable, and compliant with regulatory standards, tracing directly contributes to improved patient outcomes and more efficient healthcare services. The ability to detect and respond to security threats in real time enhances the trust that healthcare providers, patients, and regulators place in CMDs, leading to broader adoption and integration of these devices in clinical care. This means safer, more reliable care for patients, particularly those who rely on CMDs for critical monitoring or treatment. Whether a hospital-based device that tracks vital signs in an intensive care unit or a wearable sensor that monitors glucose levels in a diabetic patient, the security and functionality provided by tracing technologies ensure that these devices can be trusted to perform their roles effectively. From a healthcare delivery perspective, tracing technologies support the scalability and efficiency of digital health solutions. As healthcare systems continue to evolve and integrate more connected devices, tracing will play an essential role in managing the security and operational reliability of CMDs at scale. Tracing helps healthcare providers manage complex systems while reducing the risks associated with device malfunctions, data breaches, or regulatory non-compliance by providing the tools to monitor large numbers of devices in real time.

4 Future Developments or Challenges

As the healthcare industry continues to adopt and integrate CMDs into patient care, the role of tracing technologies will become even more crucial. However, with the evolving landscape of medical technology and cybersecurity threats, several key developments and challenges will shape the future of tracing solutions for CMDs. These factors will influence how devices are monitored and the broader implications for patient safety, regulatory compliance, and the operational efficiency of healthcare systems.

One of the most significant challenges facing the future of CMD security and tracing is the increasing complexity of devices and their growing interconnectivity within healthcare networks. As CMDs become more advanced, with new features and capabilities added, they are expected to handle more sensitive data and perform more complex tasks. Devices will no longer operate in isolation but will instead be part of an interconnected network that includes hospital systems, cloud platforms, wearable technologies, and even smart home devices. This interconnectivity raises new concerns for tracing technologies. Tracing solutions must evolve to monitor the internal operations of individual devices and their interactions with external systems. The potential for cross-device vulnerabilities increases as CMDs communicate with a broader range of devices and networks. A breach in one device could serve as an entry point for attacks on the

entire network, making it critical that tracing technologies provide visibility into the device's external communications and interactions with other systems. Additionally, as CMDs become more complex, the volume of data generated by tracing technologies will increase significantly. This data must be managed effectively to avoid overwhelming healthcare providers and security teams. Future developments in tracing will need to focus on intelligent data management, including advanced filtering and prioritization techniques, to ensure that only the most critical events are highlighted for further investigation. Moreover, tracing tools must integrate more deeply with *Security Information and Event Management* (SIEM) systems to provide healthcare organizations with a comprehensive view of their device ecosystems.

4.1 Resource Constraints in Low-End Devices

While high-end CMDs will benefit from increasing computational power and more advanced tracing capabilities, low-end devices will continue to face resource constraints that limit the types of tracing technologies that can be deployed. Wearables and other small, low-power devices are becoming more prevalent in healthcare, providing continuous monitoring of patients in remote settings. These devices typically operate on limited processing power, memory, and battery life, making it difficult to implement continuous or resource-intensive tracing. The challenge for future developments in low-end CMD tracing will be to create ultra-lightweight tracing solutions that can provide effective monitoring without consuming significant device resources. This may involve the development of more efficient algorithms for event-driven tracing, where monitoring is only triggered by specific events or conditions rather than running continuously. Additionally, advancements in edge computing may allow some of the computational burden associated with tracing to be offloaded from the device to nearby processing nodes, such as smartphones or home hubs, thereby extending the device's operational life without compromising security. Another promising avenue for development is adaptive tracing, where the tracing level can be dynamically adjusted based on the device's current workload, energy levels, or operational status. For example, a low-end, fully charged CMD that does not actively monitor the patient could run more intensive tracing activities. In contrast, during periods of high demand or low battery, the tracing could be scaled back to essential activities only.

4.2 Integration of AI and Machine Learning in Tracing

One of the most exciting developments in the future of CMD tracing is integrating AI and *Machine Learning* (ML) into tracing solutions. AI and ML algorithms can analyze vast amounts of data generated by tracing activities and identifying patterns, anomalies, and potential security threats that may be too subtle or complex for traditional detection methods. These technologies will play a critical role in predictive analysis, enabling healthcare providers and device manufacturers to identify potential vulnerabilities before they are exploited or device malfunctions before they impact patient care. AI-driven tracing solutions will allow real-time anomaly detection, providing more proactive security and operational monitoring. For example, AI algorithms could learn the normal operational patterns of a CMD and then flag any deviations from these patterns that may indicate a security breach or malfunction. These algorithms could also prioritize the most critical events, reducing the workload on security teams and ensuring that threats are addressed promptly.

Using AI and ML in tracing will also facilitate automated incident response. In the event of a detected anomaly or breach, AI-driven tracing solutions could automatically take predefined actions, such as isolating the affected device from the network, rolling back software to a secure version, or alerting the security team for further investigation. This level of automation will be crucial as the number of CMDs deployed in healthcare continues to grow, making manual monitoring and response increasingly impractical.

4.3 Regulatory and Ethical Challenges

As tracing technologies become more sophisticated and integrated into CMDs, they will also face increasing scrutiny from regulatory bodies and ethical considerations. Tracing technologies must operate within a framework that balances security and privacy. While the continuous monitoring of CMDs is essential for detecting security threats, it also raises concerns about the privacy of the device user (often the patient) and the healthcare providers interacting with these devices. Regulatory bodies such as the U.S. *Food and Drug Administration* (FDA) and the *European Union Agency for Cybersecurity* (ENISA) are likely to impose stricter guidelines on how tracing data is collected, stored, and used. These guidelines must address the need for transparency in tracing activities, ensuring that device manufacturers and healthcare providers know what data is being collected and how it is being used. This is particularly important in light of regulations like the GDPR, which imposes strict requirements on the handling of personal data. Furthermore, the ethical implications of tracing must be considered, particularly in scenarios where CMDs are used in vulnerable populations, such as elderly patients or individuals with chronic conditions. Future developments in tracing will need to ensure that patient autonomy and informed consent are maintained, allowing patients to control how their data is monitored and shared. The design of tracing systems must consider the ethical responsibility to protect patient privacy while providing the security necessary to ensure device safety and functionality.

4.4 Scalability and the Future of Connected Healthcare

As CMDs proliferate across healthcare systems worldwide, scalability will become a significant challenge for tracing technologies. Healthcare providers are already deploying increasing numbers of CMDs in hospitals, clinics, and patient homes, and this trend is only expected to accelerate with the rise of personalized medicine and remote care. Tracing technologies will need to scale to monitor vast networks of devices without overwhelming security teams or healthcare infrastructure. The future of connected healthcare, often called Healthcare 4.0, envisions a world where CMDs are integrated into large-scale, interconnected systems that communicate seamlessly across hospital networks, cloud platforms, and even smart city infrastructures. In this environment, tracing solutions will need to evolve to provide end-to-end visibility into the behavior of CMDs within these complex ecosystems. This will require advancements in distributed tracing, where data from multiple CMDs across different locations and platforms are collected, correlated, and analyzed to provide a comprehensive picture of the healthcare system's overall security and performance. Additionally, future developments in blockchain or *Distributed Ledger Technologies* (DLT) could enhance tracing by providing immutable, verifiable records of device activity, further ensuring the integrity of CMDs within interconnected healthcare systems.

5 Conclusion

As CMDs become integral to modern healthcare, the need for robust security measures, such as tracing technologies, has never been more critical. This chapter has highlighted the indispensable role of tracing in ensuring the security, privacy, and operational integrity of CMDs, particularly as these devices evolve in complexity and capability. Tracing provides early detection of anomalies by continuously monitoring device behavior, safeguarding patient safety and the broader healthcare ecosystem from potential cyber threats. For high-end CMDs, advanced tracing mechanisms like eBPF offer deep system insights and comprehensive security monitoring without compromising device performance. Conversely, low-end CMDs face unique challenges due to resource constraints, requiring lightweight, efficient tracing solutions that balance security with operational efficiency. Both categories benefit from tailored tracing strategies that enable proactive security management and regulatory compliance. Looking ahead, integrating AI and machine learning into tracing technologies will enable more intelligent and predictive security

measures, enhancing real-time threat detection and automated response. As the healthcare landscape continues to evolve, CMDs will play an increasingly central role in personalized care, making it essential that tracing technologies adapt to new threats and emerging healthcare needs. In conclusion, tracing technologies will continue to be pivotal in maintaining the trustworthiness of CMDs in an ever-connected healthcare environment. Tracing supports the future of digital healthcare by ensuring that these devices remain secure, reliable, and compliant, promoting safer and more efficient patient care across diverse healthcare settings.

Acknowledgments

This research has been funded from the European Union's research and innovation programme ENTRUST, under grant agreement No. 101095634.

References

[1] B. L. Iantovics, "The CMDS Medical Diagnosis System," in *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2007.

[2] P. A. Williams and A. J. Woodward, "Cybersecurity Vulnerabilities in Medical Devices: A Complex Environment and Multifaceted Problem," *Medical Devices: Evidence and Research*, 2015.

[3] S. D. Baker, J. Knudsen, and D. M. Ahmadi, "Security and Safety for Medical Devices and Hospitals," *Biomedical Instrumentation & Technology*, vol. 47, no. 3, 2013.

[4] M. Tavakolan and I. A. Faridi, "Applying an Energy-Aware Security Mechanism in Healthcare Internet of Things," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2020.

[5] S. Magnani, F. Risso, and D. Siracusa, "A Control Plane Enabling Automated and Fully Adaptive Network Traffic Monitoring with eBPF," *IEEE Access*, vol. 10, 2022.

[6] A. N. Godunov and F. N. Chemerev, "Tracing Tools for «Baget» Family RTOS," *Proceedings of the Institute for System Programming of the RAS*, vol. 31, no. 4, 2019.