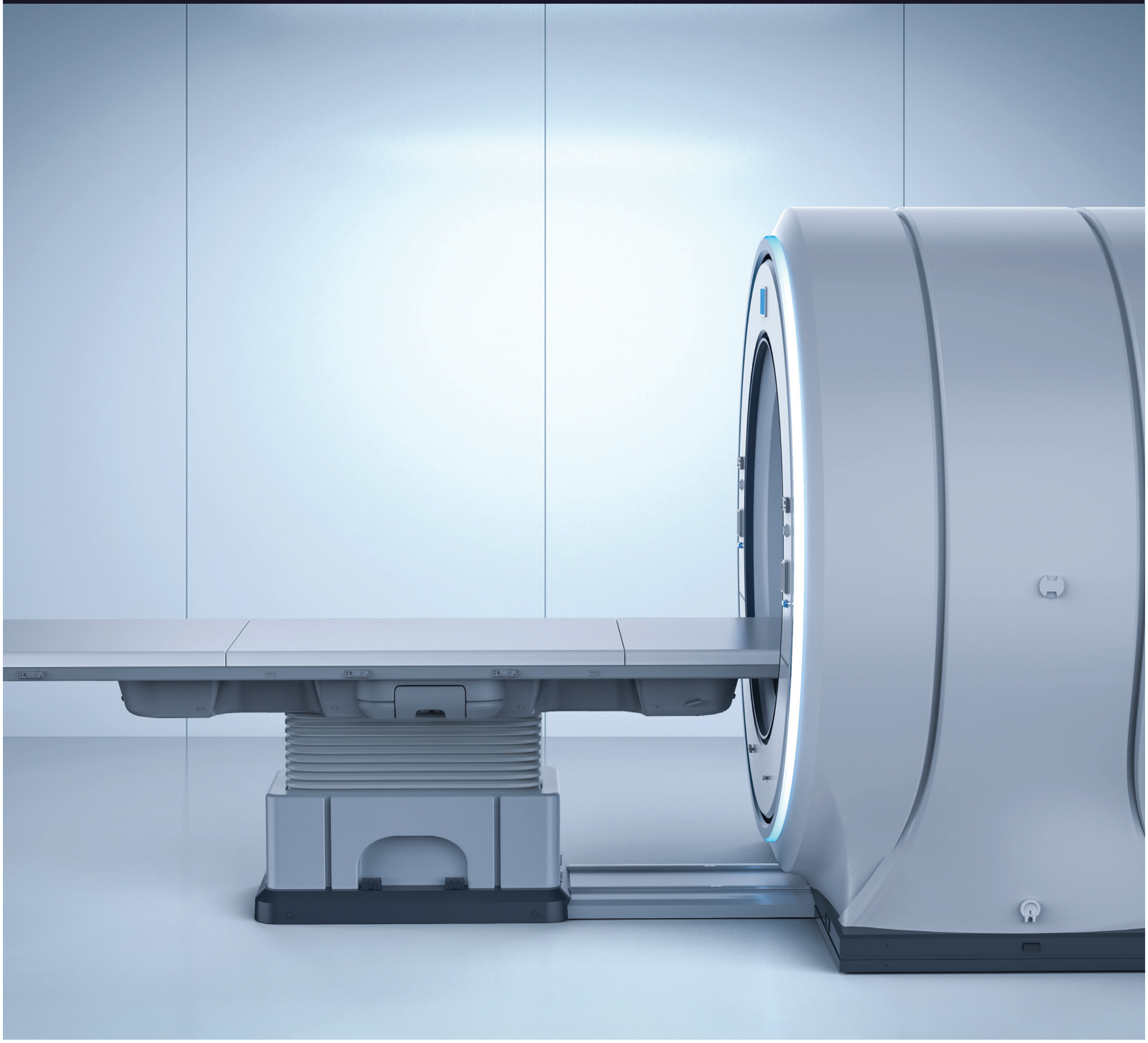
WHITEPAPER

Medical Device Safety and Security: **Obvious but not easy to achieve**

Yi Zheng



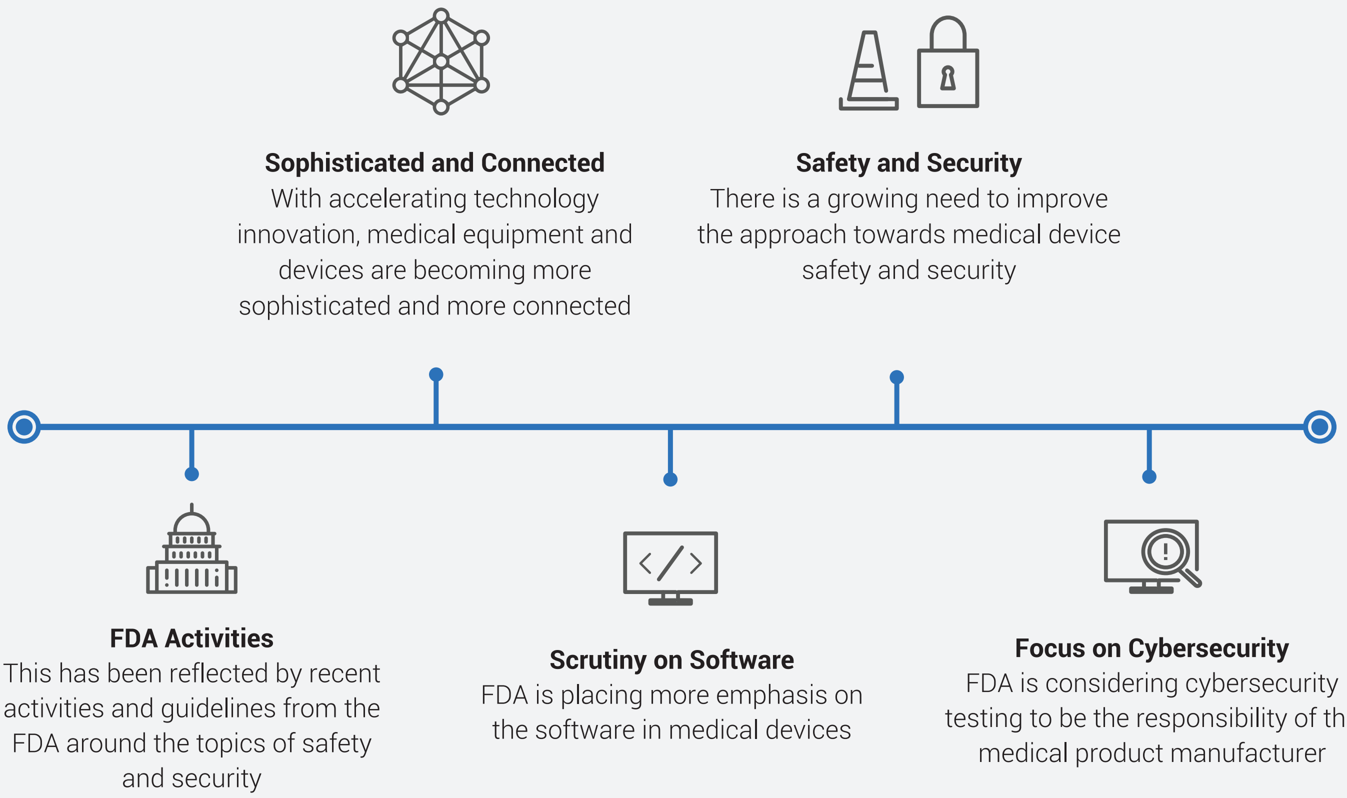
Abstract

With accelerating technology innovation, medical devices are becoming more sophisticated and more connected. Consequently, as in most mission-critical systems, safety and security are two increasingly vital requirements in medical device design. In April 2018, the U.S. Food & Drug Administration (FDA) announced plans to increase regulatory powers over medical device safety, including cybersecurity. This paper will explore challenges around safety and security requirements related to medical devices, as well as specific measures to deal with these challenges.

Introduction

Safety and security may sound like two simple words, but these two factors are essential requirements for mission-critical systems such as medical devices. Taking care of these two "simple words" in an embedded system is a lot more complex and involves more effort than what most people would think. In today's world, device manufacturers have a larger role—as an integrator of various technologies—than an engineering house that builds everything from scratch. The adequacy of safety and security needs to be gauged for specific components in the system and the end applications.

Trends for the **Medical Market**



Sophisticated and Connected
With accelerating technology innovation, medical equipment and devices are becoming more sophisticated and more connected

Safety and Security
There is a growing need to improve the approach towards medical device safety and security

FDA Activities
This has been reflected by recent activities and guidelines from the FDA around the topics of safety and security

Scrutiny on Software
FDA is placing more emphasis on the software in medical devices

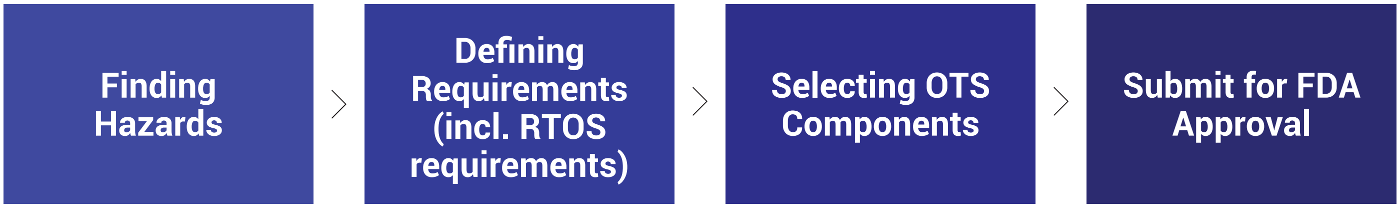
Focus on Cybersecurity
FDA is considering cybersecurity testing to be the responsibility of the medical product manufacturer

The U.S. Food and Drug Administration (FDA) regulates over 190,000 different devices manufactured by more than 18,000 firms in more than 21,000 medical device facilities worldwide. Recent changes made by the FDA have brought software to the forefront of medical device regulation. As connectivity becomes a standard feature for medical devices, security requirements are also surfacing. The FDA's recent activities reflect the market's awareness of the lurking risks. The FDA requires device makers to have a clear inventory of the software used in the device through a "Software Bill of Materials" which would include software developed by the device makers as well as that obtained Off-The-Shelf (OTS software). The plan puts emphasis on security as it considers cybersecurity testing to be the responsibility of the medical product manufacturer. It also clarifies that the medical device manufacturer will choose what software to use, thus bearing responsibility for the security as well as the safe and effective performance of the medical device. This puts a huge burden on medical device companies to not only add mission-critical security to their products, but also manage lifecycle security over the life of the device in the field.

Safety

Most device makers have a well-defined approach to functional safety. This approach is reflected in the device maker's corporate culture and internal processes. Although the FDA has provided some guidelines for safety, for example, *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices* and *General Principles of Software Validation*, etc., the onus of ensuring the safety of devices still falls upon the manufacturers. So, where does one start?

Addressing safety for medical devices follows 4 main phases:



Finding Hazards

Defining Requirements (incl. RTOS requirements)

Selecting OTS Components

Submit for FDA Approval

Safety – Finding Hazards:
Start with the Identification of Safety Hazards in the System

If we examine the multiple international standards on functional safety, we will see that many of them start with hazard identification for a safety-critical system. As a matter of fact, ISO 14971, which is titled *Medical Devices – Application of risk management to medical devices*, is an FDA-recognized standard for hazard and risk management. If we look at the system holistically, we can try to identify all the system-level hazards.

Examples of system-level hazards:

- *The user interface of the device does not accept input from the user*
- *The sensor on the device malfunctions and reports wrong data*
- *The device does not perform the prescribed action A within time T after receiving the command*

We can see how these three examples of malfunctions can cause a safety issue under certain usage scenarios. Let us take, for example, "*the user interface of the device does not accept input from the user.*" If the device is designed to administer a certain medication to the user, and if it has an interface for the user to stop the medicine, then it would be critical that this function does not break down.

If the Malfunction can Impact the Safety of the User, then it is a Safety Hazard.

You may be wondering about the difference between a safety hazard and a general malfunction of the system. The answer is quite simple: if the malfunction can impact the safety of the user, then it is a safety hazard.

If all software components were developed by the device-maker in-house, then the hazard identification process would be straight-forward. However, many of today's medical devices are sophisticated and contain off-the-shelf hardware and software products. Hazards in these components can lead to a system-level hazard. The same exercise can be applied to the other components in the system, including hardware, middleware, and any other piece that the device maker procures for the system.

Safety – Defining Requirements

Once the hazards have been identified, we must find a way to mitigate the risks that could result from those hazards. This process usually **starts with the definition of functional safety requirements**. Some of the hazards can be taken care of at the system level. For example, the risk: "*a logical error could occur in action A,*" could be handled by the following functional safety requirement: *the algorithm used to accomplish action A must be validated to ensure there is no logical error*. Assuming the device maker is designing the software for action A, they could probably use design verification method to perform the required validation for the algorithm. In this case, the requirement does not cascade down to other parts of the system.

In many cases, safety requirements do not stop at the system level. Let's use another risk from the above example hazard, "*action A must be free from interference from another action in the system*." A proper breakdown of this hazard will lead to safety requirements in the hardware, the BSP (board support package), the operating system, the various middleware modules that interact with each other and the top-level application, as well as the communication and collaboration among all these pieces.

Let's see how one system level risk can translate into safety requirements for various components within that system.

Using the example:
Device does not perform the prescribed action A within time T after receiving the command

Risk: The hardware malfunctions after receiving the command
Safety Requirement: The hardware's power unit must have failure probability lower than <threshold>

Risk: The operating system does not respond in time within time T after receiving the command
Safety Requirement: The operating system must have an upper bound for the response time less than T

Risk: A logical error could occur in action A
Safety Requirement: The design of action A must be free from logical errors

Risk: Another action B could interfere with the proper execution of action A
Safety Requirement: Action A must be free from interference from another action in the system

Safety – RTOS Requirements

Let's examine one specific case of safety requirements that follow the example used for *Defining Requirements*.

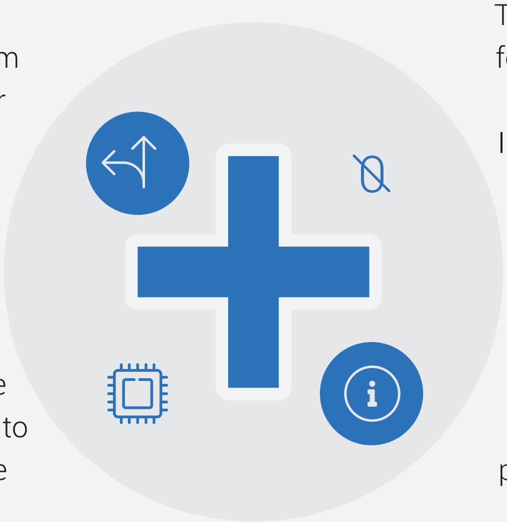
The operating system will need to have some mechanisms through which this "freedom from interference" can be achieved. In figure 1 (on page 6), we illustrate a few of the requirements. These requirements are taken from the Hazard and Risk Analysis performed on the QNX OS for Medical (QOSM) product. If you use a different operating system, you may end up having requirements that look different from these but achieve the same end goal.

This freedom from interference is perhaps one of the most crucial yet challenging aspects for a safety critical system that requires specific technical features for realization. These include well-known ones, such as microkernel architecture and temporal partitioning, as well as newer favorites such as IOMMU, a scheme that brings the MMU-like mechanism to IO devices with direct memory access. Such IO devices include graphic processing units, USB and audio devices.

How the OS can address the safety requirement defined for the RTOS: *Action A must be free from interference from another action in the system*

Run-time Isolation
The code comprising the QOSM SHALL be isolated at run time from faults, errors and failures in other components of the operating system.

IOMMU
The application guidance provided for engineers deploying the QOSM SHALL require the use of IOMMU-equivalent constraints on any hardware capable of direct memory access.



Memory Protection
The QOSM SHALL prevent code running as an application writing to memory used exclusively by the QOSM.

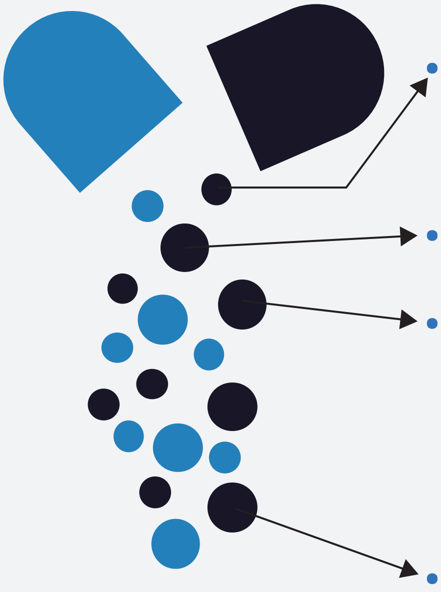
System Call Robustness
The QOSM SHALL protect itself against improper system calls. In particular it must be impossible for any application code to cause a failure in the QOSM by accidentally or maliciously passing a particular value in the parameter of a system call.

Figure 1

Safety – Selecting OTS Components

OTS is a Necessity
With the increasing complexity of today's medical devices, the use of OTS (off-the-shelf) components is a necessity

Safety Pedigree
When choosing an OTS component, it is important to understand its safety pedigree



How to select OTS components

- The supplier's past track record in similar type of systems
- Product information
- Standards compliance – the FDA has multiple recognized consensus standards for safety and security, which are listed in their guidelines for the industry
- Supplier Audit

Figure 2

We can see how top-level safety requirements can cascade down. An obvious question now is, "how do the device-makers find assurance that these safety requirements are met in an OTS component?". The answer is found in the diligence applied to the selection process.

How to Select OTS Components:

1. **A Company's Track Record**
   In many mission-critical fields, whether an Off-The-Shelf (OTS) component has been previously deployed in similar types of systems is often used as selection criteria. If there has been strong adoption for the component in the past, it says something about its reliability.

2. **Available Information for the Component**
   This information can include both marketing material that the supplier publishes on its website or through its sales channels, and technical data and specifications that can be publicly accessed. Usually not all the product information is accessible prior to procuring the OTS component. However, it is important to understand what you will get when you buy the product. For example, the functional safety requirements that are fulfilled by the OTS component may not be public information. But, do they exist? Has the supplier considered functional safety when building their product? A close look at public product information can yield important clues.

3. **Standards Compliance**
   There are several safety and security standards that are recognized by the FDA and referenced in their guidelines to the industry. As a matter of fact, the proper governance of OTS component(s) must be important enough for the FDA to issue a dedicated guideline on the topic such as: *Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software Use in Medical Devices*. IEC 62304 is one of the leading recognized consensus standards for the software safety lifecycle and is something that will be introduced further in this paper. If a supplier shows compliance to a functional safety standard, it is a good indication that the OTS component has been built with safety in mind.

4. **Supplier Audit**
   Probably the most thorough diligence is an audit of the supplier. This may not always be possible as not all companies support this type of engagement. However, this is a common practice that mission-critical industries carry out regularly with their suppliers. This type of activity can cover many grounds, from the product lifecycle to actual product artefacts. It can provide an in-depth look at how the OTS component is developed – giving you the deepest level of insight into your selection.

Safety – BlackBerry QNX Expertise

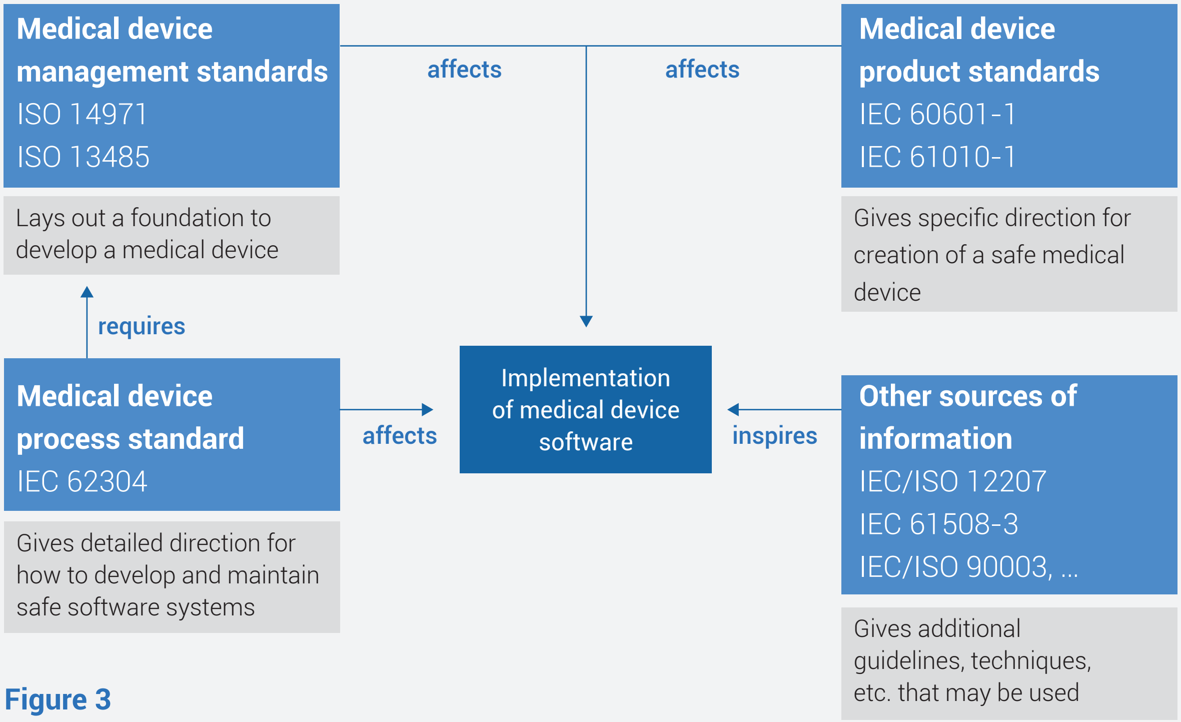
BlackBerry QNX has developed in-depth safety expertise and technology through more than 35 years of industry experience. Our functional safety DNA has been validated through the adoption of BlackBerry QNX safety certified RTOS in hundreds of millions of devices, including medical products. With safety certifications for mission critical applications being one of our core competencies, we have been helping our customers with their product development and commercialization plans not just through the use of our products, but also safety-certification consulting and professional services.

BlackBerry

- Select the right software platform—deterministic, POSIX-based, microkernel OS
- Meet reliability and safety compliance requirements
- Secure embedded software over its deployed lifecycle
- Bring products to market quicker, on budget, with quality

Safety – Standards Compliance

One of the easiest criteria to set for the OTS supplier is standard compliance. As mentioned earlier, there is no lack of guidance from the FDA for standard compliance. See figure 3 for an illustration of the various standards for quality and safety, and their relationships (taken from the IEC 62304 standard).



Medical device management standards
ISO 14971
ISO 13485

Lays out a foundation to develop a medical device

Medical device product standards
IEC 60601-1
IEC 61010-1

Gives specific direction for creation of a safe medical device

Medical device process standard
IEC 62304

Gives detailed direction for how to develop and maintain safe software systems

Implementation of medical device software

Other sources of information
IEC/ISO 12207
IEC 61508-3
IEC/ISO 90003, ...

Gives additional guidelines, techniques, etc. that may be used

Figure 3

The IEC 62304 standard defines the life cycle requirements for medical device software. The set of processes, activities, and tasks described in this standard establishes a common framework for medical device software life cycle processes. As stated by the FDA, this standard is relevant to medical devices and is recognized for its scientific and technical merit. It is also known for its support of existing regulatory policies. If a component claim complies to the IEC 62304, then you can safely assume that its development has followed a well-established safety lifecycle. Other standards feed into IEC 62304, such as ISO 14971, which defines the practice of hazard and risk analysis.

Having a requirement for OTS components can save a lot of time and effort during the due diligence stage. It can also ensure a reasonable threshold for the quality and fitness of the component you will be receiving.

Security

Security – A Growing Concern

A more recent focal point for medical devices is security, or cybersecurity, as some would prefer to call it. As software plays a bigger role in the function of the device, and connectivity becomes a standard feature for medical devices, security requirements are surfacing more and more. The FDA's recent activities reflects the market's awareness of the lurking risks.

On April 17, 2018, the United States Food and Drug Administration (FDA), announced plans to ask Congress for more funding and regulatory powers to improve its approach towards medical device safety, including cybersecurity. The FDA plans to require medical device makers to create a document called "Software Bill of Materials" for each medical device. This document would include software-related details for each product. It also wants device makers to include mandatory software update delivery systems to deliver critical security patches. The idea is to help device owners "better manage their networked assets and be aware of which devices in their inventory (or use) may be subject to vulnerabilities."

The FDA, which is the only federal government agency responsible for the cybersecurity of medical devices, now considers medical device manufacturers responsible for the validation of all software design changes. This includes any computer software changes that are required to address cybersecurity vulnerabilities. This plan declares that the FDA will not conduct pre-market security testing for medical products, as it considers cybersecurity testing to be the responsibility of the medical product manufacturer. It also clarifies that the medical device manufacturer will choose what software to use, thus bearing responsibility for the security as well as the safe and effective performance of the medical device.

While the FDA already has a set of cybersecurity guidelines, malicious attacks will likely grow as more medical equipment becomes connected and, therefore, vulnerable. The FDA cybersecurity guidelines look to mitigate serious threats to connected medical devices, especially attacks that could disrupt the operation of critical monitors and drug delivery equipment.

Security – A Challenging Topic

Since security is a newer topic than safety for medical devices, it has not enjoyed the same accumulation of knowledge, expertise and solution sets over the years. Although IT security is well known to most people, security for embedded devices calls for a different set of skills and solutions. For example, there is no clear consensus on a single dominating security standard.

However, if you searched on the topic of security solutions, there is no shortage of answers. These answers range from various models for a secure product lifecycle, to a myriad of tools for scanning product security issues, databases with hundreds of thousands of security vulnerabilities, to a wide range of security services.

Unfortunately, for someone who is relatively new to security, this flood of information is overwhelming and difficult to guide a real action plan.

Security – BlackBerry Expertise

BlackBerry has developed in-depth security expertise and technology through 30 years of industry experience. Our security DNA has been validated through the adoption of BlackBerry devices in segments with top security demands, including all 7 of the G7 governments and all 10 of the world's largest commercial banks. The 37,000+ patents and 80+ security certificates reflect the deeply-entrenched security culture at BlackBerry. Beyond its secure devices and a wide range of security services, BlackBerry QNX augments the offering with expertise in embedded security.

7/7 of the G7 governments

37K+ patents

10/10 of the world's largest commercial banks

80+ security certificates

Security – Strategy & Tactics

Let's explore several widely-accepted strategies and tactics for security. Some of these are long-term initiatives while others are meant to address immediate gaps. Chances are, every company will need a different mix of the two to reach sufficient levels of security.

List of Strategy & Tactics

- Review your security posture
- Identify vulnerabilities in deployed products
- Build protection mechanisms in current-generation products
- Build a culture of security as the foundation
- Leverage external expertise where possible
- Adopt and adapt recognized process models in the market
- Architect your next generation product with best in class security technologies and practices and, using defense in depth topologies
- Manage the life cycle of security for the life of the product in the field

First, if security has not been a focal point before, a good understanding of your security posture is needed. This type of assessment is best performed by security experts, ideally someone with domain knowledge for your industry. This activity takes stock of existing security mechanisms in place and identifies the major gaps in your internal process and product.

Despite the differences between safety and security, one parallel that could be drawn between the two disciplines is that they require focus on both internal process and specific product definition. Of all the items listed under "strategy and tactics," *build a culture of security as the foundation* is probably the most esoteric-sounding one. How do you "build" a culture?

Internal processes certainly play an important role. Although there are university courses on subjects such as encryption, the truth is we don't graduate from university readily prepared to build security into our first product. Also, one person's approach to security may be quite different from another. A well-defined internal process for security is the necessary foundation for a security culture. Although some standards try to prescribe what this process should look like, such as ISO 21434, a truly effective process must suit the idiosyncrasies of the company, sincerely endorsed by its executives and embraced by its people.

The internal process could define higher-level stuff like the complete security lifecycle, to finer-granularity stuff such as the rules around vulnerability assessment, or the tools and methods used to scan products for security issues. It takes time for an internal process to become a natural part of everybody's work life – that is what it takes to build a culture. If a good security culture is in place, then many of these items will just follow naturally. For example, *identify vulnerabilities in deployed products* – a good internal process would have called out vulnerability assessment as a mandatory verification & validation (V&V) method. Or, "build protection mechanism" would be a basic requirement for any security lifecycle. The two last items on this list are just reminders to leverage external help, whether it is through security consulting professionals or existing security standards, to put into place a working culture.

Now that we have covered the internal process, we can look at security specification for the product. Recall for functional safety, we started with system level hazard and then examined each component to determine which component-level hazards could lead to the system-level hazard. This approach does not exactly apply to security.

What security risks exist for a medical device? Suppose a malicious hacker gains control of an infusion pump. The hacker may interfere with the normal operation of the device; such as administering a dose of the medication, or not administering any medication when the user prompts it to. The various places in the system where an attacker could "get in" to do evil work are called attack surfaces. If we draw a parallel with functional safety, these attack surfaces are like hazards. When you analyze your system for security, the goal is to identify where vulnerable places are. The operating system is an attack surface, because the operating system is the central brain. If a hacker can gain access to the operating system, say, with root privileges, then they basically have carte blanche to do whatever they please.

Security – Protecting the OS

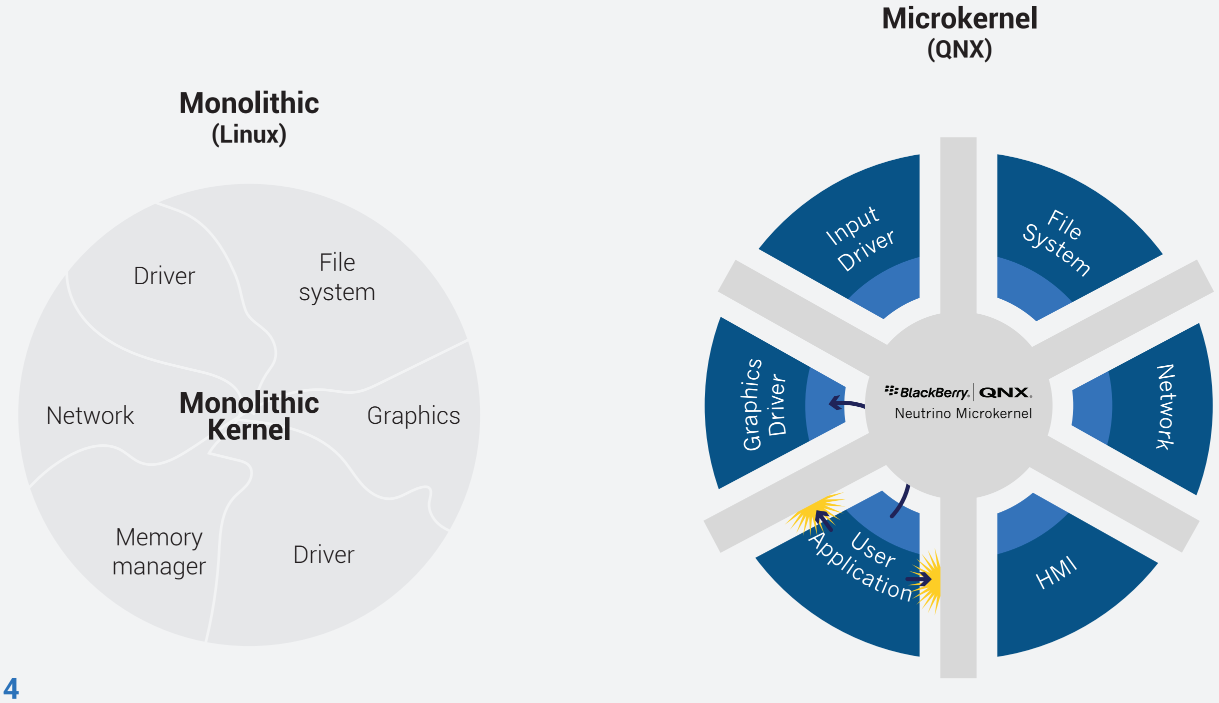


Figure 4

What do we do next once we have identified the attack surfaces? Just like in safety, security must be applied at each component level.

How do we protect against these loopholes? Well, again, you will need to have a good understanding of these components and figure out what renders them vulnerable. Let's use the operating system again. Fortunately, the QNX operating system has architecture that is naturally more resilient than a monolithic operating system. In a monolithic operating system, everything resides in one large space. Critical components like the kernel and the drivers share the same memory pool with each other. This provides a convenient route for hackers to get in from any component that is a bit vulnerable and into the brain of the system.

Once the hacker gets control of something critical, say the instruction pointer, he or she is free to run any code they wish to exploit the system. In a microkernel system, the kernel **does not** share the memory space with other components, including drivers. Everything runs in its own dedicated memory space and cannot use the memory space of another component. This makes the hacker's life much more difficult. It is important to note that the same architecture is also very useful for functional safety as it provides freedom from interference, which is a key mechanism required to build systems of mixed safety criticality.

Security – Software Solution

Apart from inherent architecture, there are solutions that can be applied to these components to make them more secure. We will use the example of Process Manager Abilities to illustrate this.

Example: Process Manager Abilities

Fine-grained control of process privileges

65+

Over 65 abilities to control various aspects of privileged operations

Abilities can be granted, denied, locked or inherited

Root and on-root abilities

Abilities control a process's ability to perform certain operations. For those of you who are not familiar with BlackBerry QNX, a process is the smallest container unit in our microkernel operating system, which contains one or more threads that actually run and "do work."

You have probably heard of the command "fork." The fork () command creates a new process. The new process (child process) is an exact copy of the calling process (parent process), with a few differences such as process ID, number of threads, timer values, etc. There is an ability governing the usage of fork. If you are using a central security policy that only gives fork privileges to certain trusted processes, it can prevent against ploys such as a fork attack, a denial of service (DoS) attack in which the fork system call is recursively used until all system resources execute a command. The system eventually becomes overloaded and is unable to respond to any input. Another example is the ability to map physical memory. This ability basically allows the process to invoke mmap to map a memory region into a process's address space. The QNX microkernel OS comes with a memory management unit, which hides the physical memory from most of the processes in the operating system but presents them with virtual memories. This alone is a very secure approach however, sometimes a process needs to use physical memory, such as in the case of a process belonging to a hardware device driver. Using a security policy, you can restrict which processes can map physical address and mitigate security risks. As you can see, the abilities are defined in fine granularity. There are over 65 different abilities you can grant, deny, lock or inherit.

Summary/Conclusion

Safety and security may sound like two simple words, but these two factors are essential requirements for mission-critical medical equipment and other devices. After reading this whitepaper, you can begin to understand that taking care of safety and security is necessary in maintaining healthy embedded systems.

Furthermore, this whitepaper can be summarized into four main points:

1. Safety and security are two vital areas to focus on for medical devices
2. It is the device maker's responsibility to ensure safety and security
3. Scrutiny of OTS components is necessary
4. Safety and Security expertise is essential for making the right decisions at the right time

About BlackBerry QNX

BlackBerry QNX, is a leading supplier of safe, secure, and trusted operating systems, development tools, and professional services for connected embedded systems. Global leaders such as Ford, Audi, Cisco, General Electric, Lockheed Martin, and Siemens depend on BlackBerry QNX technologies for their next generation of secure vehicle software platforms, network routers, medical devices, industrial automation systems, security and defense systems, and other mission and/or life-critical applications. This includes full software lifecycle management via secure over the air software updates. Founded in 1980, BlackBerry QNX is headquartered in Ottawa, Canada, with its products distributed in over 100 countries worldwide.

BlackBerry. | QNX.