Connected healthcare

Ricardo Camacho and **Mark Pitchford** look at the pivotal role of medical devices in the world of connected healthcare

Connectivity throughout healthcare is yielding huge benefits for the industry and patients alike, and is a trend set to continue and accelerate. However, cybersecurity is an issue and while the need for a secure enterprise-level architecture is widely acknowledged, the role played by securely coded devices is easier to ignore, yet vitally important.

While patients and providers benefit from improved operational efficiency derived from the use of real-time data from a wide range of sources there is, however, a downside.

As the number of medical devices networked increases, so does the number of different points (“attack vectors”) accessible to any bad actor looking to manipulate data and cause mischief.

In 2011, the ethical hacker Barnaby Jack shone a spotlight on the issue by using modified antennae and software to demonstrate how it was possible to wirelessly attack, and take control of, Medtronic’s implantable insulin pumps and to command them to administer a fatal dose of insulin.

More recent examples demonstrate that such a direct attack remains a challenge. On August 23, 2017, for example, the Food and Drug Administration (FDA) in the US approved a firmware update to reduce the risk of patient harm due to potential exploitation of cybersecurity vulnerabilities for certain Abbott (formerly St. Jude Medical) pacemakers.

The WannaCry malware attack on the UK’s National Health Service (NHS) is another example from 2017. The malware exploited the Windows implementation of the Server Message Block (SMB) protocol to propagate, targeting MRI and CT scanners, which ran on XP workstations. These medical devices were encrypted and held for ransom, which prevented safe and effective patient treatment.

The attack was estimated to have affected more than 200,000 computers across 150 countries, with estimates of total damages ranging from hundreds of millions to billions of dollars.

Defence in depth

The diversity in the nature of attacks illustrates why no single defensive measure can ever solve the problem.

There needs to be basic housekeeping such as updating older operating systems, securing protocols, and updating and validating software and firmware. But even with those precautions, there are countless ways to attack a system and an attacker only needs a single vulnerability.

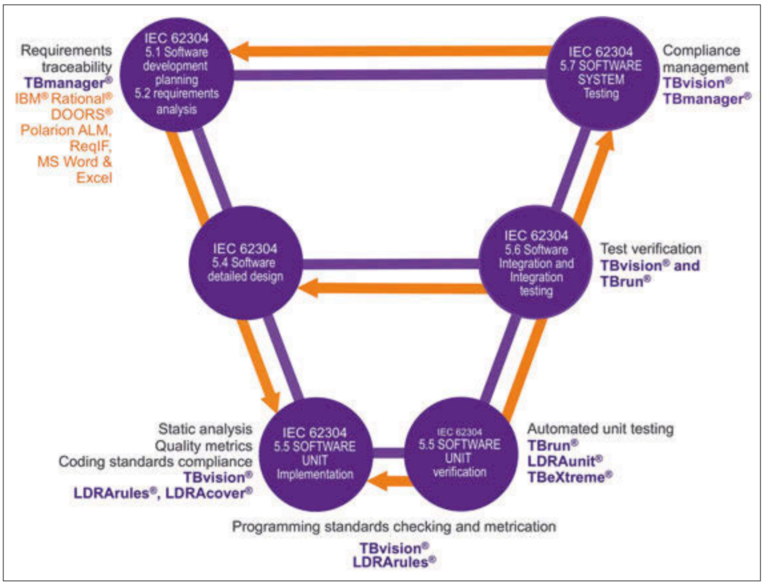
Requirements traceability
TBmanager®
IBM® Rational® DOORS®
Polarion ALM,
ReqIF,
MS Word & Excel
IEC 62304 5.1 Software development planning 5.2 requirements analysis
IEC 62304 5.7 SOFTWARE SYSTEM Testing
Compliance management
TBvision®
TBmanager®
IEC 62304 5.4 Software detailed design
IEC 62304 5.6 Software Integration and Integration testing
Test verification
TBvision® and TBrun®
Static analysis
Quality metrics
Coding standards compliance
TBvision®
LDRArules®, LDRAcover®
IEC 62304 5.5 SOFTWARE UNIT Implementation
IEC 62304 5.5 SOFTWARE UNIT verification
Automated unit testing
TBrun®
LDRAunit®
TBeXtreme®
Programming standards checking and metrication
TBvision®
LDRArules®

Figure 1: Mapping the capabilities of the LDRA tool suite to the guidelines of IEC 62304:2006 +AMD1:2015

According to Professor James Reason many aspects of medical endeavour require human input, and the inevitable human error that goes with it. But generally, there are so many levels of defence that for a catastrophe to happen, an entire sequence of failures is required.

Reason likened this to a sequence of slices of ‘Swiss Cheese’, except that in his model the holes in the ‘slices’ are forever moving, closing, widening and shrinking.

Just like the checks and controls applicable to human input into medical systems, a multiple-level approach to cybersecurity makes a great deal of sense, such that if aggressors get past the first line of defence, then there are others in waiting.

Approaches and technologies that can contribute to these defences include secure network architectures, data encryption, secure middleware, and domain separation.

The medical devices deserve particular attention, however. For an aggressor, the infrastructure surrounding them is a means to an end and only the devices themselves provide the means to threaten.

Medical devices and cybersecurity

In the past, embedded medical software has usually been for static, fixed function, device specific applications. Isolation was sufficient guarantee of security. The approach to cybersecurity and secure software development tended to be reactive: develop the software, and then use penetration, fuzz, and functional test to expose any weaknesses.

The practice of “patching” to address weaknesses found in the field is essentially an extension to this principle, but device manufacturers have a poor track record of delivering patches in a timely fashion.

In implicit acknowledgment of that situation, in October 2018 the MITRE Corporation and the FDA released their "Medical Device Cybersecurity" playbook consisting of four phases – preparation; detection and analysis; containment, eradication and recovery; and post-incident recovery.

Adopting a proactive approach to cybersecurity in medical devices "Preparation" is perhaps the key element to take from this incident response cycle – not only in identifying the security measures that are in place for existing devices, but in proactively designing them into new products.

One approach to designing in cybersecurity is to mirror the development processes advocated by functional-safety standards such as IEC 62304 'Medical device software – software life cycle processes'.

The IEC 62304 provides a common framework to develop software that full-fills requirements of addressing quality, risk and software safety throughout all aspects of the software development lifecycle.

Using a structured development lifecycle in this way not only applies best practices to the development lifecycle, it also creates a traceable collection of artefacts that are invaluable in helping to provide a quick response should a breach occur.

Beyond the safety implications of any breach, this approach addresses the FDA recommendation that any medical device must not allow sensitive data from being viewed or accessed by an unauthorised entity. The data must remain protected and accurate, preventing hackers from altering a diagnosis or important patient information.

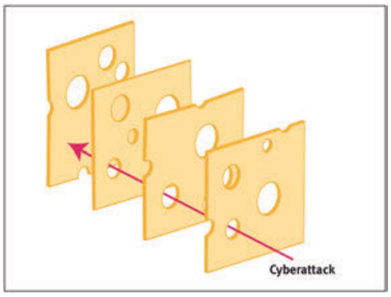

Figure 2: The 'Swiss Cheese' Model, illustrating how a sequence of imperfect defensive layers will only fail when those imperfections coincide

Secure code development

Compliance with the processes advocated by regulations can be demonstrated most efficiently by applying automated tools.

Although there are some differences between developing functionally safe and cybersecure applications, there are many similarities too. For example, both perspectives benefit from the definition of appropriate requirements at the outset, and from the bidirectional traceability of those requirements to make sure that they are completely implemented.

Unit testing and dynamic analysis is equally applicable to both functional safety and cybersecurity too, and in the latter case is vital to ensure (for example) that defence mechanisms are effective, and that there is no vulnerability to attack where boundary values are applied.

IEC 62304 also requires the use of coding standards to restrict the use of the specified programming language to a safe subset. In practice, code written to be functionally safe is generally also secure because the same malpractices in programming language application often give rise to both safety and security concerns.

Conclusions

No connected medical system is ever going to be both useful and absolutely impenetrable. It makes sense to protect it proportionately to the level of risk involved if it were to be compromised, and that means applying multiple levels of security.

Medical devices themselves deserve particular attention because they provide the primary means to threaten. The structure development approach of a functional safety standard such as IEC 62304 can provide the ideal framework to apply a proactive approach to the development of a secure applications.

Happily, many of the most appropriate quality assurance techniques for secure coding are well proven in the field of functional safety. These techniques include static analysis to ensure the appropriate application of coding standards, dynamic code coverage analysis to check for any excess "rogue code", and the tracing of requirements throughout the development process.

The legacy of such a development process includes a structured set of artefacts that provide an ideal reference should a breach of security occur in the field. Given the dynamic nature of the endless battle between hackers and solution providers, optimising breach response times is not merely a good idea. It is a potential lifesaver.

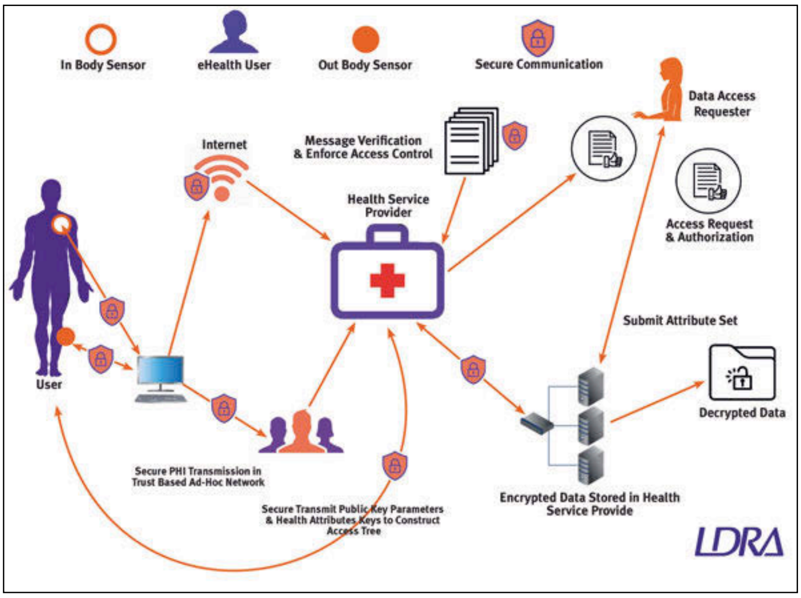
In Body Sensor
eHealth User
Out Body Sensor
Secure Communication
Data Access Requester
Internet
Message Verification & Enforce Access Control
Health Service Provider
Access Request & Authorization
User
Submit Attribute Set
Decrypted Data
Secure PHI Transmission in Trust Based Ad-Hoc Network
Secure Transmit Public Key Parameters & Health Attributes Keys to Construct Access Tree
Encrypted Data Stored in Health Service Provide
LDRA

Figure 3: The size and complexity of a typical health delivery network presents a large attack surface to potential bad actors