medcrypt

BENEFITING FROM SOFTWARE TRANSPARENCY

From SBOM to Vulnerability Management

Software bills of material (SBOM) capture software used in products. SBOMs are prerequisites to proactive product security, as well as vulnerability and risk management programs. However, extracting the full potential value of SBOMs at scale will take sustained effort, requiring tooling to overcome inherent complexities.

INTRODUCTION

Over the last decade, healthcare has seen a technology-enabled transformation of how technology enhances how care is planned, delivered, and sustained. The transformative nature of medical technology and connectivity allows for advancements by leveraging data to deliver clinical value, enabling wireless monitoring, and reducing costs through remote support.

The value of this digital transformation is only as strong as the proactive cybersecurity posture built into it. And we are seeing a trend shifting security from reactive to proactive. Medical devices are being designed with security in mind. Manufacturers are building more robust security programs. We're in the middle of a cybersecurity transformation in healthcare.

While we transform, cybersecurity challenges are growing. For example, in recent years, we've seen dozens of cases of widespread and deeply embedded vulnerabilities, like Urgent/11 or Amnesia:33. These have the potential to impact the delivery of care to patients, yet healthcare (i.e., hospitals, manufacturers, regulators) were unable to quickly answer the question "which devices are affected?"

Vulnerabilities are expected in all forms of technology. Ideally, "Every device would meet a security baseline; every device would be easily updatable; and patients would receive timely updates." (FDA, 2019). However, updating medical devices is not that simple. First, hospitals have to determine whether they have affected devices. For that to happen, device manufacturers have to know which versions of their devices are impacted and communicate that knowledge, allowing hospitals to identify at-risk assets, and then implement the resolution (assuming an action can be taken/has been tested for a device).

To do so requires a number of new processes and information sets which, in turn, will require the use of tools and automation so that complexity and scalability barriers can be overcome. The focus of this white paper is on the Software Bill of Materials (SBOM), a vital ingredient for healthcare's transformation in vulnerability management.

HEALTHCARE USE CASE FOR THE SBOM

In 1994 the FDA introduced the food label into our daily lives, ensuring "ingredients" are clearly identified. This knowledge of ingredients can lead to informed decisions by consumers, underscoring the FDA philosophy to advocate for transparency across its various domains. It's thus no surprise that this concept has persisted to the disclosure of software "ingredients" included in medical devices.

In technology, the SBOM is analogous to food labels. An SBOM is essentially a listing of components in a piece of software that uniquely identifies each component, including version, and other relevant descriptors where applicable. In the U.S., FDA has signaled its plan to require SBOMs and timely patching from all manufacturers with requests for budget and new regulatory authorities.

The value of knowing the components in a device was perhaps most broadly felt with the release of some of the pervasive vulnerabilities of the recent past (e.g., QNX BrakTooth). As regulators, MDMs, and HDOs all struggled to determine whether they had been impacted, and identify affected devices, they found that the versions of software deployed were not regularly captured, nor documented in a usable/searchable way. And given the vulnerability impacted a pervasively deployed component, the scope of examination required felt limitless and resulted in a suboptimal response.

If SBOMs were universally available, accurate, and healthcare knew how to use them, vulnerability management would be much easier. But we are just at the beginning of the process of using the software ingredient list to make vulnerability management easier. Right now, it's hard; we're in transition.

EVOLUTION OF THINKING

Implementing the full range of functionality required for an effective vulnerability management program, and thus fulfilling the promise of better cybersecurity, is not trivial and will need to address both systemic and idiosyncratic challenges.

Systemically, traditional engineering tools lack the ability to address the entire range of security process requirements, premarket through postmarket. Providing good security management throughout the devices' lifecycle means that processes and tools need to provide a rich feature set, as well as support a range of use cases.

For example, developers of medical device software should have tools available to continuously flag vulnerabilities in their software stack and recommendations for how to dispose of those vulnerabilities before the software goes through QA, much less shipped. We need universally available and accurate SBOMs for that, plus tooling to automate much of the workflow.

Further, we need to recognize complexity as part of the challenge. This is on several levels, starting with the SBOM itself and the number of software components, identified vulnerabilities, depth of analysis required, and identification of dependencies between components. This directly leads to challenges of version and change management as well as scalability, especially for large, multi-platform manufacturers.

CASE STUDY: WHAT'S IN A NAME?

Although standards and conventions for software component naming have emerged, they are not specific enough to allow for automated processes to uniquely and definitely identify individual components, resulting in naming and versioning inconsistencies that require resolution.

A recently released vulnerability related to BlackBerry's QNX RTOS tells a complicated story (see table on right). Originally the supplier was identified as QNX, then once acquired was noted as BlackBerry, but only for a period of time. Additionally, in parsing names a similar name was used for multiple products, which can lead to confusion in identifying what is actually implemented, and thus potential vulnerabilities that apply.

Additionally, products can have different names in different languages and platforms. For example, the ZMQ transport library can be called libzmq, zeromq, or cppzmq. But it requires a developer to point out that cppzmq is a binding for the ZMQ library, and would need further investigation to understand if the same vulnerabilities impact it.

One must be able to resolve the inconsistency in component definition and naming as well as versioning, leading to the need of eliminating duplicates, resolving non-matches, and identifying missed components - in other words, reduce the number of false positives and false negatives.

DEPENDENCY NAME	DEPENDENCY SUPPLIER	DEPENDENCY VERSION	UPDATE
neutrino rtos	qnx	6.1.0	-
qnx	qnx	4.25a	-
qnx neutrino rtos	blackberry	6.4.1	-
qnx neutrino rtos	blackberry	-	SP1
qnx neutrino rtos	blackberry	6.5.0	-
qnx neutrino rtos	blackberry	6.5.0	SP1
qnx rtos	qnx	4.25	-
rtos	qnx	6.3.0	-

CULTURE AND RESISTANCE TO CHANGE

Lastly, the benefits of providing SBOMs are not widely accepted as good business practices. Although the security benefits, as well as regulator and customer expectations are apparent, some argue there is an intellectual property concern or business risk to sharing. The recent Executive Order on Improving the Nation's Cybersecurity signals this will not be the case going forward. Between Federal procurement and FDA delaying submissions that do not meet cybersecurity requirements, there is a growing demand for this type of information to enable better cybersecurity visibility and consequently management.

This leads to the fundamental, idiosyncratic challenge that the requirement of providing an SBOM may not be appreciated as beneficial and a requirement by all stakeholders.

PRACTICALITIES

Being able to extract a multi-level SBOM, in its full depth and correctly identifying all components and dependencies, provides tangible benefits from a software and quality management perspective. It enables efficient and reliable management of software supply chain risks and is a prerequisite to proper security vulnerability identification and management.

According to both pre- and postmarket guidance documents from the FDA, reliable vulnerability prioritization and efficient response management can only be accomplished with the implementation of a comprehensive SBOM management program.

Premarket Objectives:
In the premarket scenario (i.e., during software design and development), the challenge is to provide visibility and enable mitigation of a large number of vulnerabilities across a large number of diverse software components. Those developing a product would also benefit from having full SBOM visibility to enable checks on supply chain decisions.

During design and development, the focus should be on the ability to process (i.e., identify and fix) a large number of vulnerabilities and to avoid analysis paralysis that a complete risk assessment may provide. With efficiency as the goal, it should be acceptable at this stage to address a vulnerability that may not need fixing or to miss one that would be picked up later in a more thorough risk assessment. As a new product proceeds towards market release, a more thorough and complex, risk-based analysis of the remaining vulnerabilities should be performed.

Postmarket Objectives:
In a device's postmarket life, SBOMs are a valuable tool that can be applied to identify devices and versions affected by newly uncovered vulnerabilities. In a simple example, if a new vulnerability could be exploited by a (hypothetical) WannaCry 2.0 malware, SBOM analysis provides targeted insight of which device and specific versions contain the vulnerable software component. This approach provides for a focused and targeted response, rather than assuming that most products would be at risk simply based on the fact that there isn't evidence to the contrary.

It would be shortsighted though to say the SBOM is the only requirement in a vulnerability response. Understanding the attack surface, clinical use case, implementation, exploitability, and other insights into device operation would inform how to respond to an identified vulnerability.

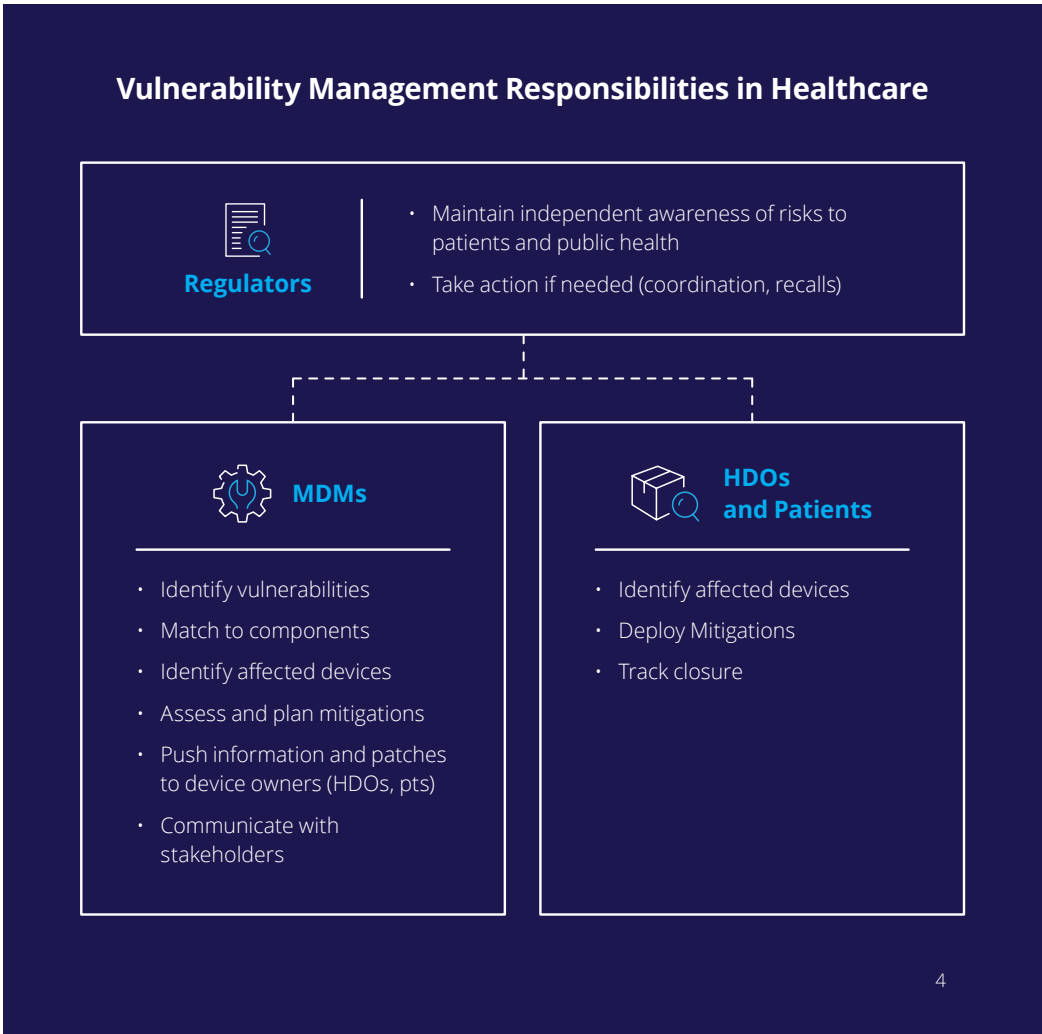
Mitigations should be deployed to the devices that have truly been affected, rather than applied broadly and unnecessarily consuming resources. In that sense, providing SBOM information to hospitals allows them to analyze their asset database and identify the affected, and only the affected, devices for mitigation (e.g., patch deployment). Considering the complexity of change management in the clinical environment, such targeted response and focus on high-priority risks is essential to a successful security management program.

SBOM EXECUTION CHALLENGES

The path to successful SBOM implementation is not an easy one. Technical and execution challenges are plenty, ranging from inconsistent software component naming to the management of the complexities of the SBOM itself, to organizational challenges such as determining which groups are responsible for vulnerability mitigation.

Taking a step back and looking at the purpose of SBOMs - the intent is not to create another regulatory requirement and yet another piece of documentation for the device's design history file. The overarching purpose is to ensure patient safety by enabling easy and efficient communication about device vulnerabilities (and by extension device's risks) between manufacturers, regulators, and operators (i.e., the HDO), while empowering each stakeholder to make informed judgements. For this to work we need to solve for the inherent complexity of vulnerability management of today's software-based medical devices.

Each device contains hundreds if not thousands of software components and each healthcare delivery organization has thousands if not ten-thousands medical devices on their network. MDMs, HDOs, and regulators each have a role in monitoring and managing this deluge of data:



For every one of the responsibilities listed in the graphic, stakeholders in the healthcare industry are challenged to execute at scale. Considering the complexities and large numbers of devices and software components on hospital networks, such processes can only work successfully through tooling and automation. Trying to accomplish something like this manually and with limited software component visibility would result in high efforts on all sides and would deliver substandard results. In other words, this might check a compliance box, but would not result in a sufficiently secure state.

MDMs, HDOs, and regulators are taking steps to develop the processes and adopt the technologies needed to fulfill vulnerability management. Soon, we expect it'll be a standardized and scalable way of doing software business in healthcare. Healthcare is on the journey but we are just not quite there yet.

CONCLUSION

Every SBOM program could become higher quality, or more efficient, even the most mature, but how do we know how much is needed, right now? Start by generating, then managing, then optimizing against the real risks (both security and business)

The responsibility to secure healthcare is shared across industry participants, which results in the need for efficient and reliable security and vulnerability communication. Cybersecurity can be improved by building it directly into medical devices and by maintaining the deployed devices' security posture. Therefore, we significantly lower the security burden placed on the ecosystem.

Security is never done, it's continuous. But with a minimum level of investment in SBOM tooling and vulnerability management process improvement, cybersecurity risks across the breadth of stakeholders (from regulator, to customer, to patient) become more manageable.

medcrypt

MedCrypt provides proactive security for healthcare technology. MedCrypt's platform brings core cybersecurity features to medical devices with just a few lines of code, ensuring devices are secure by design. MedCrypt announced a $5.3 million Series A funding round in May of 2019, bringing the total funds raised to $9.4 million with participation from Eniac Ventures, Section 32, Y Combinator, and more. The company is based in San Diego, California.

For further details, please visit and contact:

Website: www.medcrypt.com
Email: info@medcrypt.com
Twitter: @MedCrypt