GRAMMATECH

Managing Software Supply Chain Risk in Medical Devices

Modern medical devices are gaining complexity, and as connectivity to the internet, cloud, and outside world increases, so does the security challenge. Further, medical devices for home use are increasing exponentially, so devices must withstand a non-clinical environment while communicating on insecure home networks. And with medical devices, security risks are also safety risks, which increases development costs and liability.

To address functionality, cost, and time-to-market concerns, medical devices rely on third-party and in-house developed applications. Although the FDA already requires software supply chain risk management for medical device software development, risk management has often underestimated the vulnerability of third-party components due to a lack of technology to analyze and comprehend the impact of this software. To address these obstacles, medical device software developers require new development approaches, tools, and techniques.

WHAT IS SOUP?

ISO/IEC 62304, which defines a risk and quality-driven software development process for medical device software, mentions the term “software of unknown pedigree/provenance.”

SOUP refers to software that is largely unknown in terms of its development processes, verification, and validation, and may be unsupported by a company or organization.

According to IEC 62304, “SOUP," software of unknown provenance, [is a] SOFTWARE ITEM that is already developed and generally available and that has not been developed for the purpose of being incorporated in the MEDICAL DEVICE (also known as “off-the-shelf software”) or software previously developed for which adequate records of the development PROCESSES are not available.”

It also states, in section 7.1.2 (part of the risk-management process) details, “The MANUFACTURER shall identify potential causes of the SOFTWARE ITEM identified above contributing to a hazardous situation ... failure or unexpected results from SOUP.”

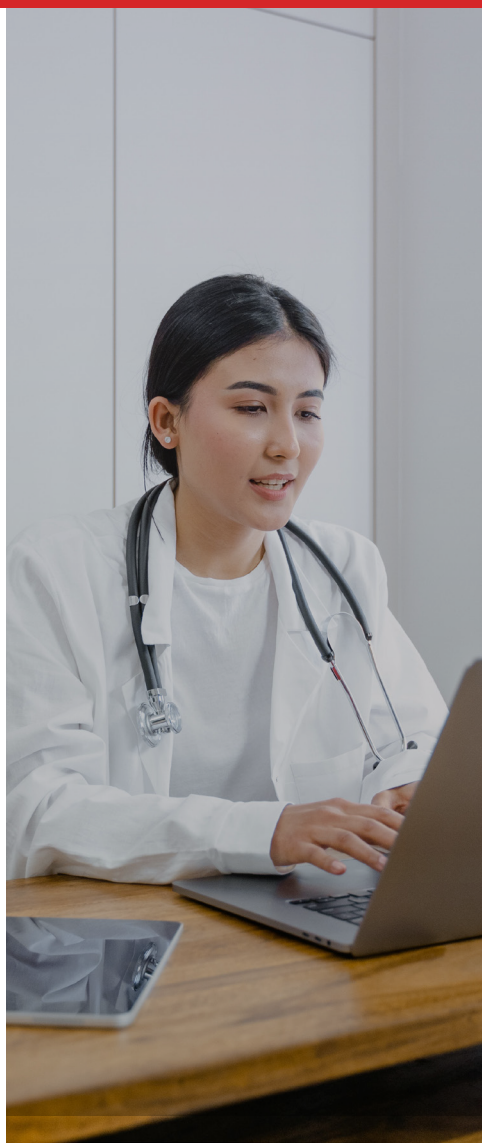
Also, in section B.1.2, Field of Application states, “It is assumed that through proper performance of MEDICAL DEVICE RISK MANAGEMENT, the MANUFACTURER would understand the component and recognize that it includes SOUP.” A medical device manufacturer needs to understand and manage the risks of using SOUP effectively. Retroactively unit testing third-party source is difficult, and for binary products it’s nearly impossible.

WHAT ABOUT OFF THE SHELF SOFTWARE?

It is important to note that not all off the shelf software is SOUP, though some can be SOUP. Software from reputable vendors who specialize in safety-critical software, such as **GrammaTech**, is not considered SOUP. Tools like **CodeSonar**, for example, are not of unknown provenance and are designed to meet safety critical standards.

Any off-the-shelf product or open source project that lacks documented development processes and standards, without as a track record of verification or validation, poses a risk for use in safety-critical medical device software.

MANAGING RISK IN SOUP



Risk management of third-party software and other SOUP is already a required activity for FDA pre-market approval of medical devices. Safety is the primary concern, but security is becoming equally important, with cyber-attacks, among other potential threats, putting safety at risk. With the increased risk from external software sources, it is critical to leverage technologies developed to analyze and fix software vulnerabilities. Software composition analysis can now provide insight into third-party code, even in binary form, which greatly aids supply chain risk management.

Security has become an increasingly important consideration, and the FDA addressed this in its 2014 guidance on the subject. Because security design, coding, and testing require a unique set of skills, they frequently fall outside the expertise of embedded software managers and developers. Furthermore, when managing third-party code, such as operating systems, libraries, or open source software, security is rarely top of mind. It takes time and money to evaluate these outside software sources. Automated tools are critical for understanding the risk and managing it. Furthermore, there must be a clear method for documenting, tracking, and communicating the security risks associated with SOUP.

THE ROLE OF THE SBOM

Tools such as **GrammaTech CodeSentry** can analyze SOUP and determine the constituent components of open source and OTS even when the only available media is binary files. In doing do, it generates an SBOM and vulnerability report which determines the risk the SOUP component poses. SBOMs also provide:

- **IDENTIFYING AND AVOIDING VULNERABILITIES** in reused components in your own developed software and software purchased by your organization is paramount.
- **MANAGING SOFTWARE SUPPLY CHAIN RISK** in order to eliminate and reduce unknown security risks in re-purposed software. SBOMs provide data for business decisions on software purchases and open source reuse.
- **SUPPLY CHAIN QUALIFICATION** to ensure consistency and accountability from suppliers. Suppliers that meet the SBOM requirements during procurement are given preferential treatment.
- **IMPROVED SECURITY AND DOWNSTREAM BENEFITS** that come with risk management and mitigation. Avoiding and catching security risks before they become embedded in their products pays huge dividends during development and deployment of your products.
- **COMMON UNDERSTANDING OF SOFTWARE ASSETS** that comes with a standardized SBOM amongst software developers, suppliers, and open source projects. SBOMs have become a way to communicate software content and dependencies within and outside an organization.

GRAMMATECH

CODESentry

Bill of Materials

Name	Version	Security Score	Match Level	Path	Number of CVEs
ben	0.6.1ubuntu2	100	Low	Demo/Example/COTS.zip/libtidy.dll	0
expat	2.0.1	100	High	Demo/Example/COTS.zip/program.exe	0
freetype	2.8.1	35	High	Demo/Example/COTS.zip/CoreGraphics.dll	1
libcue	2.2.1	100	Low	Demo/Example/COTS.zip/CoreGraphics.dll	0
libflac	1.3.2	100	Low	Demo/Example/COTS.zip/CoreAudioToolbox.dll	0
libjpeg	6b2	29	High	Demo/Example/COTS.zip/CoreGraphics.dll	2
libnvidia-gl	460.67	100	Low	Demo/Example/COTS.zip/Admin.dll	0
libnvidia-gl	460.67	100	Low	Demo/Example/COTS.zip/program.exe	0
libpng	1.2.56	2	Low	Demo/Example/COTS.zip/gnsdk_manager.dll	7
libpng	1.2.56	2	Low	Demo/Example/COTS.zip/CoreMedia.dll	7
libpng	1.2.56	2	Medium	Demo/Example/COTS.zip/CoreGraphics.dll	7
libtiff	3.9.4	9	Medium	Demo/Example/COTS.zip/CoreGraphics.dll	55
libxml	2.9.3	100	High	Demo/Example/COTS.zip/WebKit.dll	0
libxml	2.9.4	100	High	Demo/Example/COTS.zip/libxml2.dll	0
libxslt	1.1.28	2	High	Demo/Example/COTS.zip/libxslt.dll	10
nsight-systems-target	11.0.3	100	High	Demo/Example/COTS.zip/SQLite3.dll	0
nvidia-cuda-toolkit	9.1.85	100	Medium	Demo/Example/COTS.zip/gnsdk_manager.dll	0
nvidia-cuda-toolkit	5.5.22	100	Low	Demo/Example/COTS.zip/gnsdk_dsp.dll	0
nvidia-cuda-toolkit	9.1.85	100	Medium	Demo/Example/COTS.zip/zlib1.dll	0
nvidia-cuda-toolkit	9.1.85	100	Medium	Demo/Example/COTS.zip/CoreMedia.dll	0
nvidia-cuda-toolkit	9.1.85	100	Medium	Demo/Example/COTS.zip/CoreGraphics.dll	0
openssl	1.0.2u	25	High	Demo/Example/COTS.zip/Admin.dll	5
openssl	1.0.2u	25	High	Demo/Example/COTS.zip/program.exe	5
tidy	5.2.0	100	High	Demo/Example/COTS.zip/libtidy.dll	0
zlib	1.2.5	100	High	Demo/Example/COTS.zip/gnsdk_manager.dll	0
zlib	1.2.5	100	High	Demo/Example/COTS.zip/gnsdk_dsp.dll	0
zlib	1.2.11	100	High	Demo/Example/COTS.zip/zlib1.dll	0
zlib	1.2.11	100	High	Demo/Example/COTS.zip/CoreMedia.dll	0

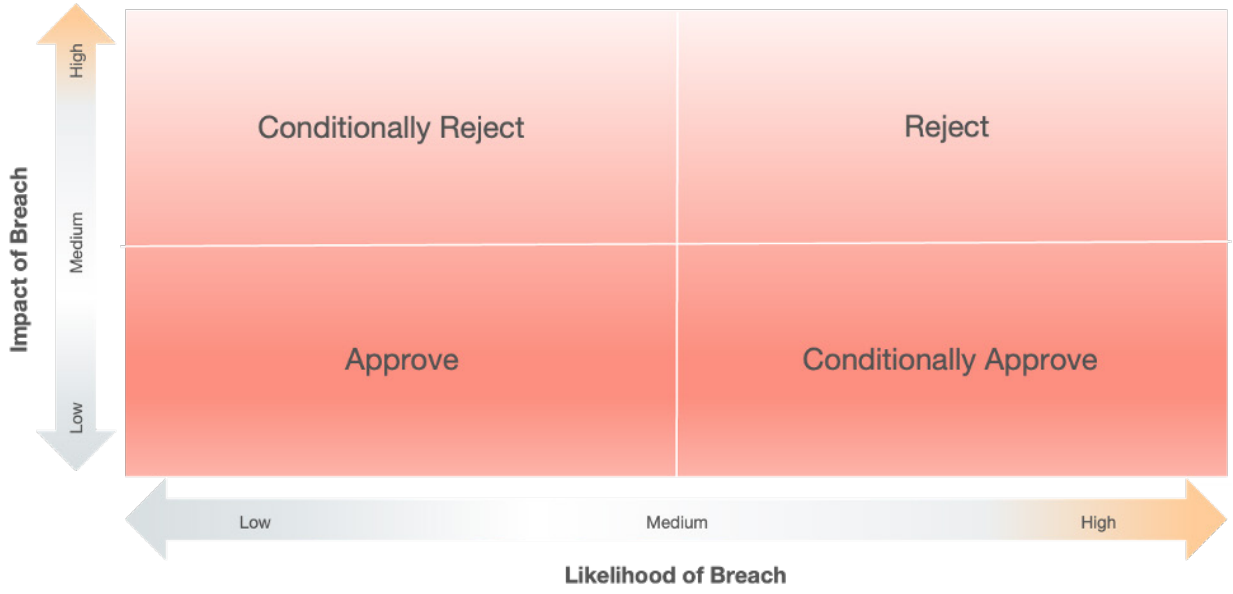
An example SBOM output from CodeSentry

SOUP AND OTS RISK MANAGEMENT

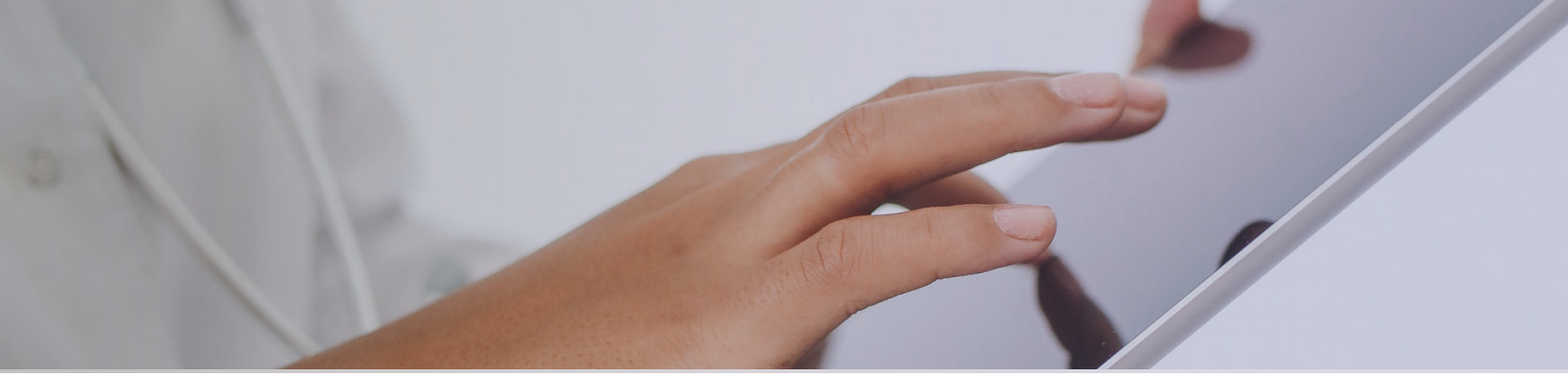
With all the data you have from using a solution like **CodeSentry** to produce a SBOM, security score, vulnerability report, license information, etc., It seems like a daunting task to figure out what to do with it and how to deal with the discovery of critical vulnerabilities.

Like most issues with security, it's important to evaluate results in terms of likelihood and impact. A vulnerability's likelihood is a determination the possibility of an attack succeeding using the discovered vulnerability. Impact is the possible damage caused by a successful attack. This impact needs to consider not just the immediate damage or exposure but also the long-term impact on your brand, bottom line, and customer experience.

A good way to evaluate the critical open source vulnerabilities found in COTS software is with the risk quadrant as shown below. For example, software with some vulnerabilities, deemed unlikely to be exploited with low impact, could be approved for purchase, renewal, or maintenance contract by simply accepting the low-risk level. Obviously, software with high impact, high likelihood of attack vulnerabilities may need to be rejected (more on that below).

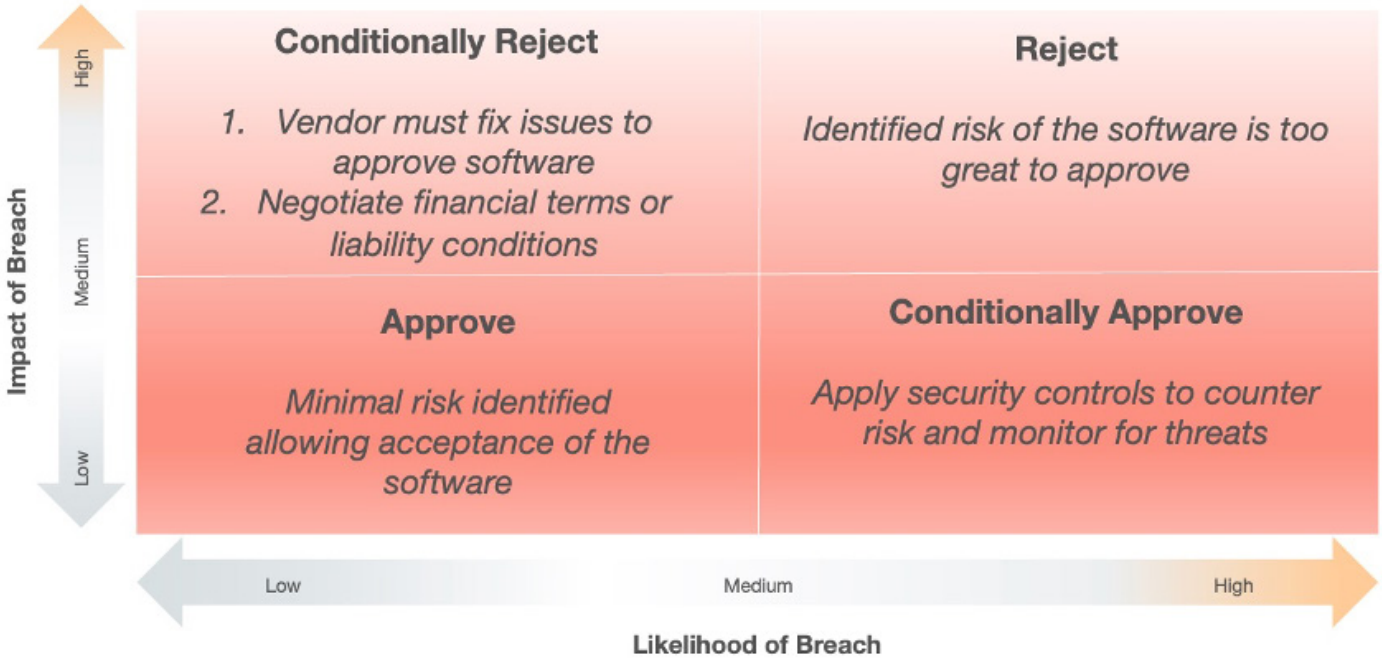


The security vulnerability risk quadrant



There is more subtlety to the decision making since it is often not possible to simply reject software that is critical to the product. This adds another dimension to the decision making and the approach taken with the software vendor: Using SBOMs in the COTS procurement process

Managing software procurement with SBOMs is relatively new territory but the assumption here is the both the user and supplier are acting in good faith with the aim to improve the security of the product and reduce the security risk over time. This thinking can be applied to software already deployed today by analyzing the most critical software and making more intelligent security decisions. The illustration below shows a more nuanced decision process to follow once SBOM results are in-hand.



Decision making process for handling SBOM results

A NEW APPROACH TO SOFTWARE SUPPLY CHAIN RISK MANAGEMENT

New approaches to medical device software development are required if current development can't keep pace with market challenges. The following approach can help lower risk and liability from third party software:

TRAINING ON AND ADOPTION OF NEW SOFTWARE DEVELOPMENT, SAFETY, AND SECURITY BEST PRACTICES AND GUIDELINES

Software development is evolving, as are techniques and methodologies for developing security-critical and safety-critical devices. A long-term plan for the adoption of incremental and iterative development, risk management, and security best practices is beneficial, both in product quality and development productivity. Including software as part of supply chain management is critical to evaluating this software for safety, quality, and security.

COMBINE RISK MANAGEMENT WITH SECURITY THREAT ANALYSIS AND ASSESSMENTS

As with treating security properly at all stages of development, security must be part of a medical device risk management plan. The scope of this assessment must include OTS and SOUP from the supply chain. An insecure device is most likely not a safe device, and a safety analysis can overlook security.

MANAGE THE SOFTWARE SUPPLY CHAIN

Most projects require reuse or building upon open source or commercial products. Assessing the quality and security of SOUP and other code, internal or external to the company, can reduce the risk.

COMBINE RISK MANAGEMENT WITH SECURITY THREAT ANALYSIS AND ASSESSMENTS

As with treating security properly at all stages of development, security must be part of a medical device risk management plan. The scope of this assessment must include OTS and SOUP from the supply chain. An insecure device is most likely not a safe device, and a safety analysis can overlook security.

INTEGRATE DEVELOPMENT AND TESTING TOOL AUTOMATION

Software development tools are advancing along with new techniques and methodologies. Binary software composition analysis such as that provided by GrammaTech CodeSentry, play an important role in automating the vetting of open source, SOUP and OTS. Along with advanced static analysis of CodeSonar, forms part of a modern tool chain that is critical in seizing the increased security, quality, and safety that new techniques and approaches offer.

INTEGRATE SECURITY AND SAFETY AUDITS INTO CI/CD

Auditing software under development on a continuous basis and ensuring quality, security, and safety at all stages is critical to success. Automation enables the ability to continually monitor the state of security of a product under development when integrated as part continuous integration/continuous deliver pipeline. Ensuring that products meet the audit standard before shipping illustrates proper due diligence and risk management required for FDA pre-market approval. When done continuously, there are no “surprises” late in development cycle.

This list seems like a tall order to adopt in the short term; fortunately, it is a long-term recommendation. For an example of what to do now, starting with a basic supply chain audit is a great way to understand your starting point.

SUMMARY

Software development success depends on smart decision making, including build-versus-buy decisions. Bringing in outside source and binary code, the SOUP, has its risks, and proper management of risk in terms of safety and security is required. However, the risk of SOUP can be managed through diligent and continuous use of software composition analysis, SBOM generation and disclosure, and integrating both into a continuous development pipeline.

GRAMMATECH