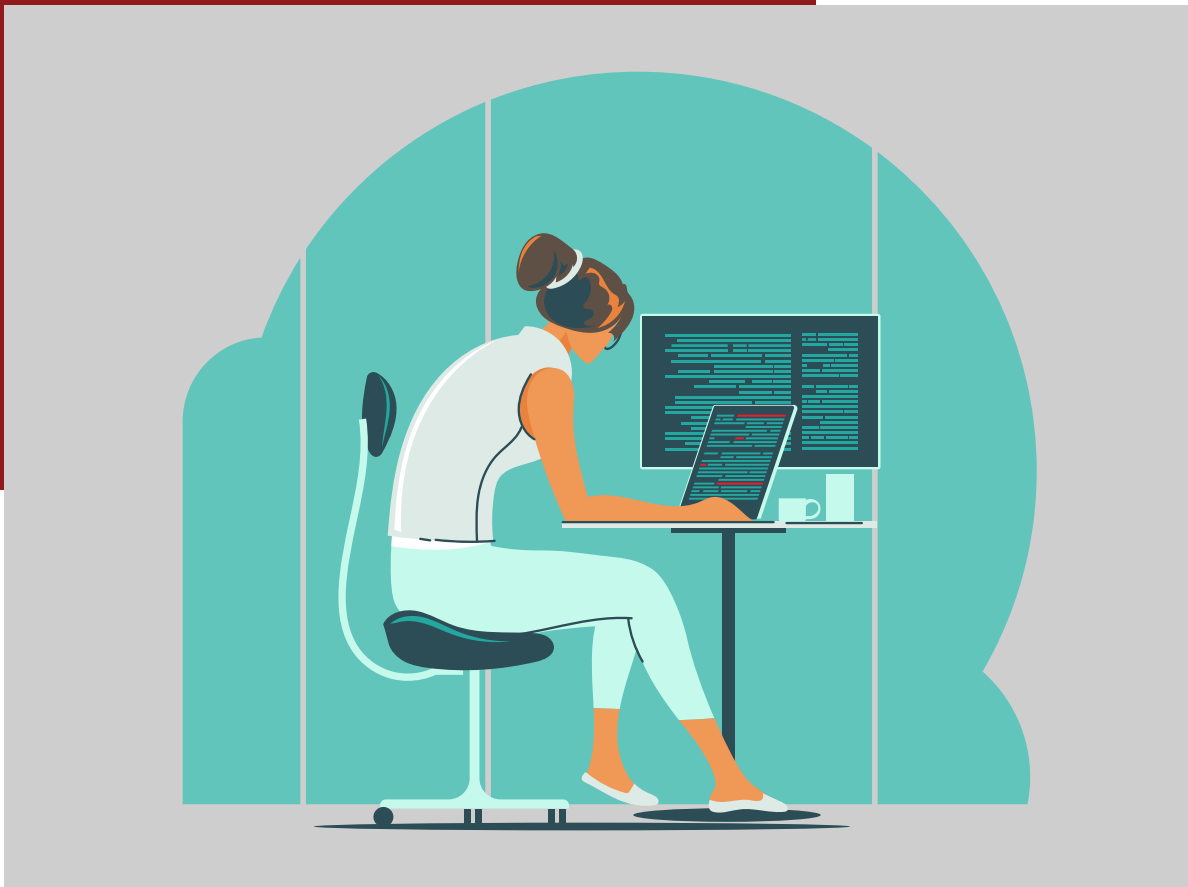
CODESECURE

Using a SBOM to Make Better Software Security Decisions



WWW.CODESECURE.COM

INTRODUCTION

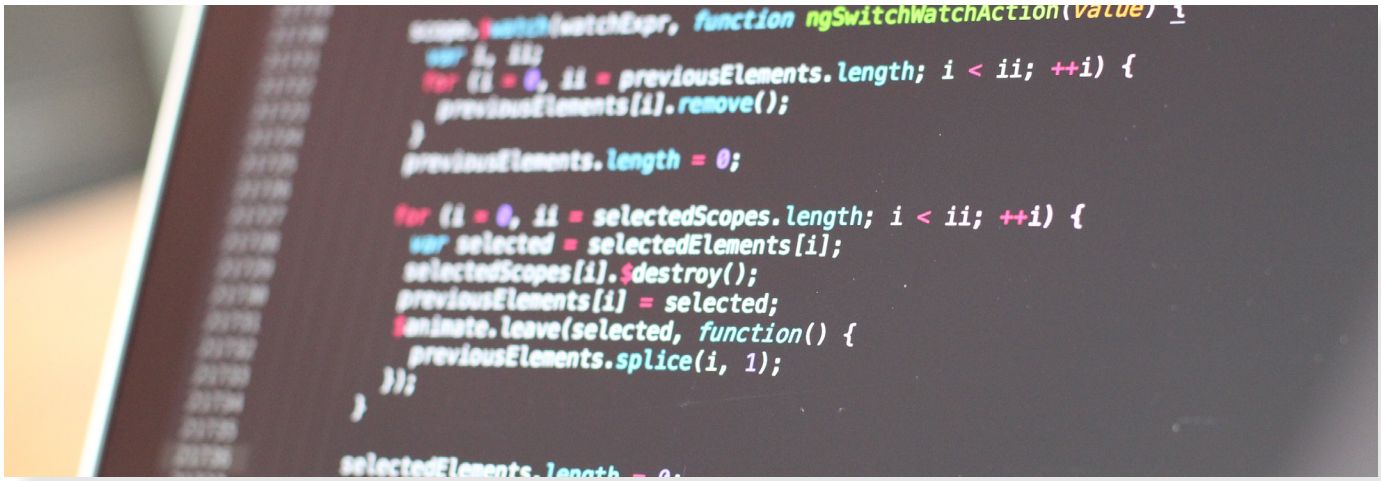
Software supply chain attacks are on the rise. Many of the high-profile cybersecurity news stories such the SolarWinds attack and the Apache Log4j vulnerability tell a tale of attackers exploiting vulnerabilities and weaknesses in the software supply chain. The mode of operation can range from fairly simple exploits of known vulnerabilities like Log4Shell to very sophisticated attacks, sponsored by nation state threat actors such as the SolarWinds supply chain attack.

The annual spend on enterprise software (also known as commercial off the shelf or COTS software) is now approaching $600 billion with a growth rate of 11.5%. Yet, given the magnitude on this investment, how much is spent on ensuring the all this COTS software is secure? In relative terms, enterprises are spending a pittance on securing their software supply chain versus their investment in software applications to support their operations. This is why vulnerable COTS software is so dangerous, as vulnerabilities can be "hidden" in open source components (Log4j) within the products, and customers are left with a false sense of security.

In a 2021 Forrester's report on the State of Application Security, a survey of companies showed that the most likely avenue of a cyberattack was software vulnerabilities in their applications. Open source software plays a key role in the prevalence of these vulnerabilities. Its use is increasing year over year and the number of vulnerabilities is increasing even faster.

Additionally, a 2021 Osterman Research Report found that 100% of widely used COTS software analyzed by CodeSecure CodeSentry (without access to source code) contained vulnerable open source components. A worrying 85% of those analyzed applications had critical vulnerabilities with a common vulnerability scoring system (CVSS) score of 10.0. The most vulnerable applications were everyday email and online meeting applications. All the applications are widely used applications and the presence of these critical vulnerabilities presents a serious cybersecurity risk to organizations as the likelihood and impact of a successful cyberattack is very high. The result of this research makes it clear that much more needs to be done to secure the software supply chain.

So, that begs the question of what can enterprises do to improve this situation?



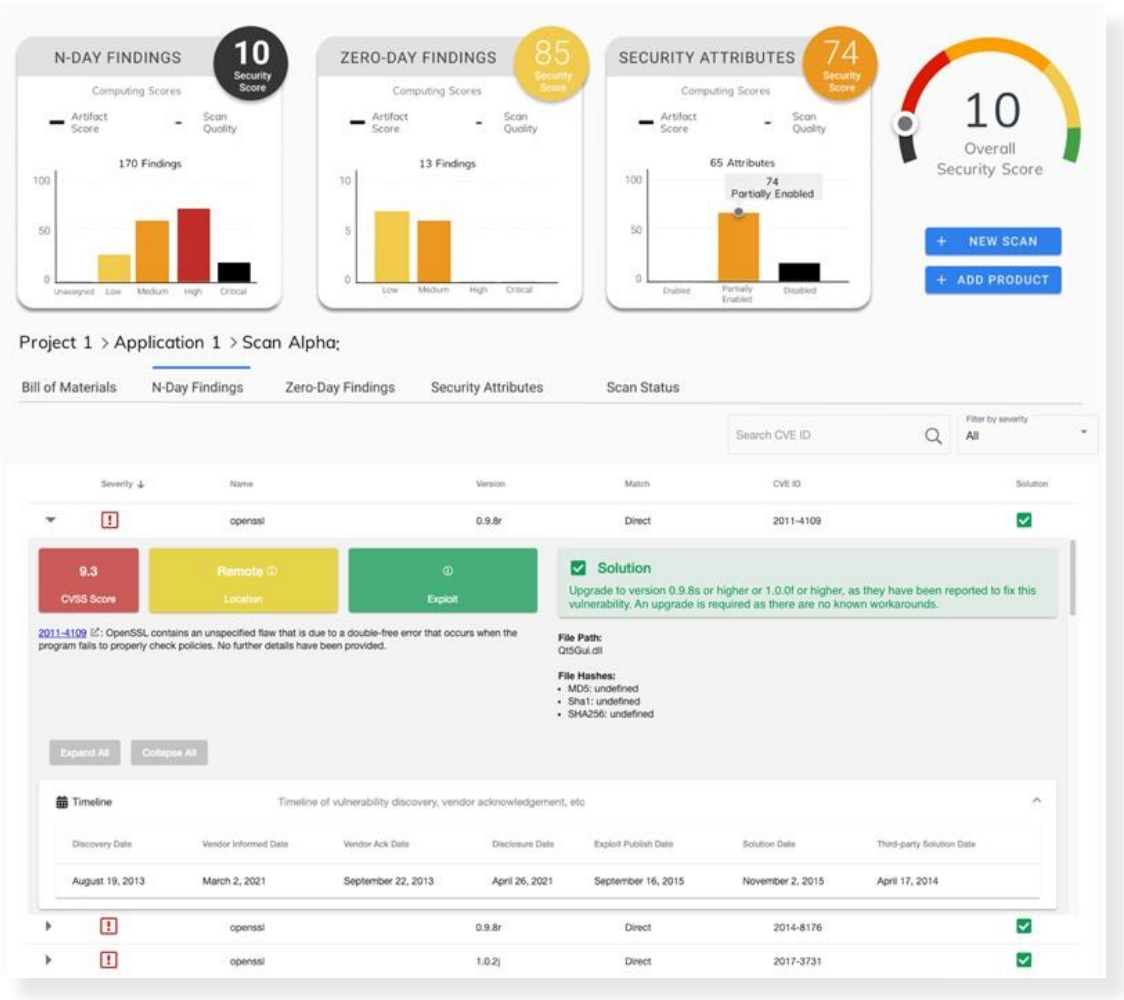
HOW CAN ENTERPRISES IMPROVE SOFTWARE SECURITY POSTURE?

The traditional approach is based on trusting each of your vendors and assuming they have done the correct due diligence during the development of their software. Formally, this would mean asking the vendor to attest to their good software engineering practices during the procurement process and disclose the security practices for supporting their software. Customers, on the other hand, are left to investigate the security of the products they use through associations or user groups in order to share information about vendor risk and software security.

These approaches are clearly not enough as evidenced by the Apache Log4j vulnerability. Despite the best intentions of software vendors, too many security vulnerabilities are lurking in third-party and open source components in COTS software. This represents a software security blind spot that the vendors themselves may not even know about. The key artifact needed to shed light on this blind spot is the software bill of materials (SBOM).

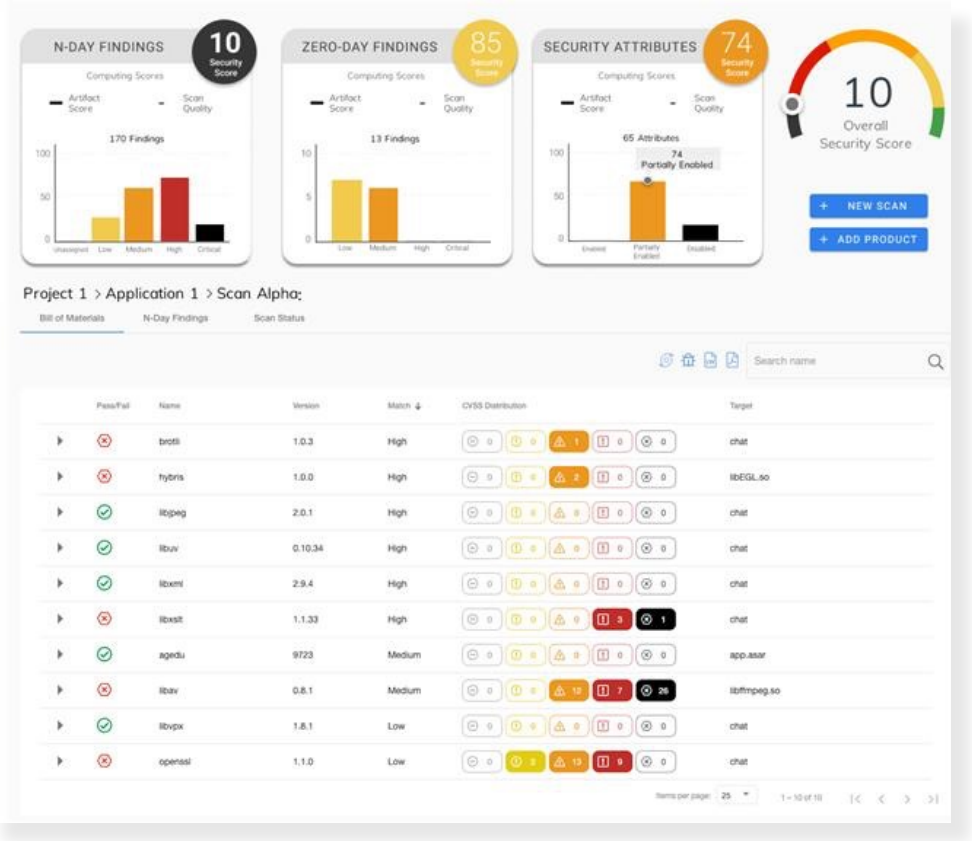
The SBOM is an inventory report of the software components that make up a software product. Just like the labels on food products, a list of ingredients and nutritional information. The SBOM is same thing: A full disclose all of the components that are built into a product. However, just as organizations are struggling to receive an SBOM from their vendor they are also struggling with what to do with an SBOM once they have it. What if the software you've procured has critical vulnerabilities? How do you deal with the information and how do you deal with your vendors?

Findings from Osterman report show that all enterprise applications use open source components and most had known critical exploitable vulnerabilities in them. It is obvious that existing perimeter security approaches won't be enough to defend organizations from threats against these software vulnerabilities. The goal of automating software supply chain security is to get deep visibility into the products you are purchasing and have deployed to support critical functions of the business. The SBOM plus detailed vulnerability information is needed to truly understand the security risk of the software within your organization. With new technology that analyzes binary applications without the need to access source code, enterprise security teams can now produce their own detailed SBOMs along high level dashboards to help analyze and summarize the findings. In addition, a software vulnerability report is critical in cataloging the known (N-day and Zero-day) vulnerabilities in the software components outlined in the SBOM. An example of a CodeSentry executive summary dashboard is shown below:



An example executive summary dashboard from CodeSentry.

The generated SBOM needs be both human and machine readable to export for disclosure/sharing with other organizations and integrate with other security and risk solutions. Human readable formats should provide easy navigation of the components and the vulnerabilities reported. Machine readable outputs such as CycloneDX, SPDX and others must conform to industry standards to support open interchange. An example of the SBOM report from CodeSentry is shown below.



An example of a detailed SBOM from CodeSentry

Looking at the example application above, it's clear there are serious vulnerabilities in the software, both high and critical severity security issues, that warrant concern about using this software in an enterprise environment.

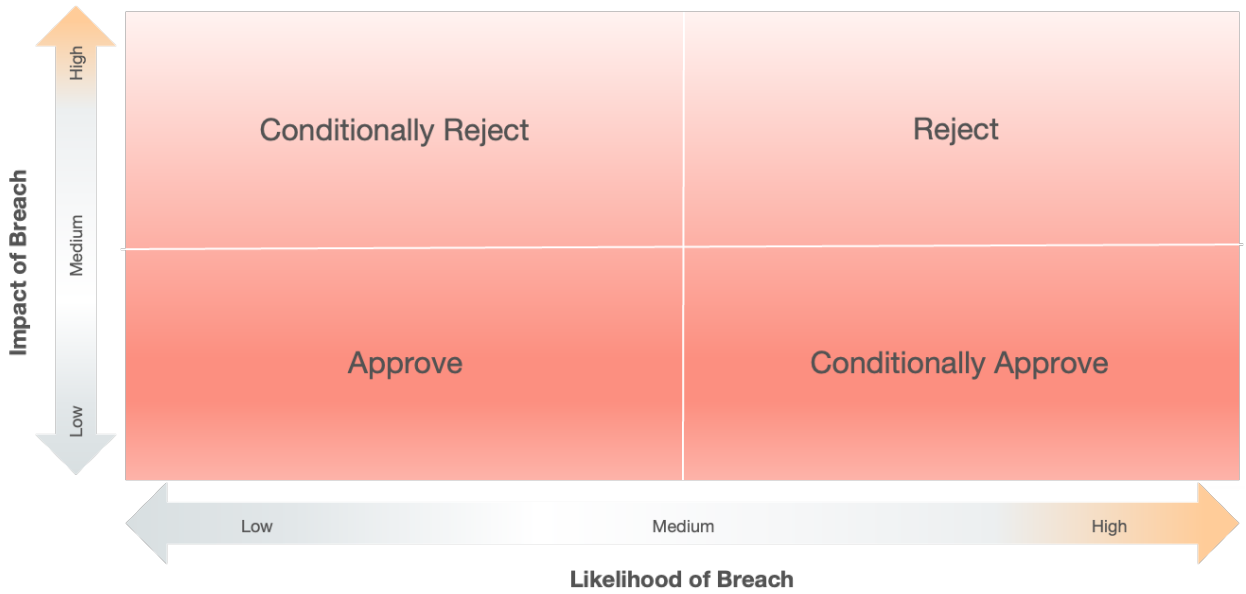
Additional information in the SBOM can include licensing. Checking license information helps ensure compliance and reduces risk that the software is released or consumed with unlicensed components. Having license information can also help with forensics when checking what version of the open source component may be vulnerable such as which version of Apache Log4j included the critical Zero-day vulnerability.

Given this information, what is the next step?

WHAT TO DO WITH SBOM RESULTS?

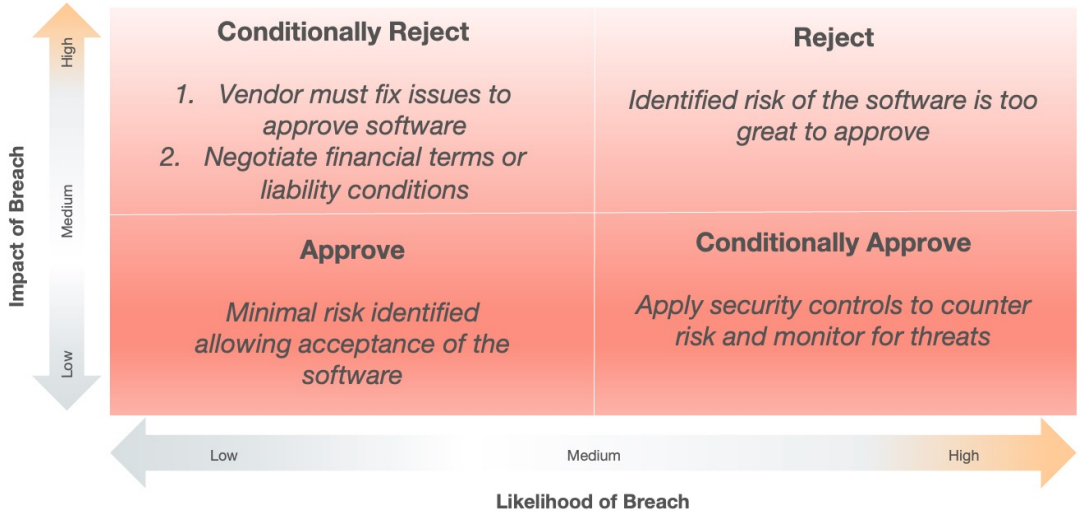
With all the data you know have from using a solution like CodeSentry to produce a SBOM, security score, vulnerability report, license information, etc., it seems like a daunting task to figure out what you do with it all and how to deal with fallout from the discovery of critical vulnerabilities. Like most issues with security, it's important to evaluate results in terms likelihood and impact. 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 to your brand, bottom line, and customer experience. Considering that, on average in the United States, companies paid $6.53 million per data breach, judging potential impact needs to be taken seriously.

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 accepted 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 your business. This adds another dimension to the decision making and the approach taken with the software vendor. Using SBOMs in the COTS procurement process is relatively new territory but the assumption here is the both the customer (you) and the vendor are going to act in good faith with the aim to improve the security of the product and reduce the security risk over time. You can also apply this thinking to software you already have deployed today by analyzing what's most critical to your business and making more intelligent security decisions. The illustration below shows a more nuanced decision process to follow once SBOM results are in-hand.



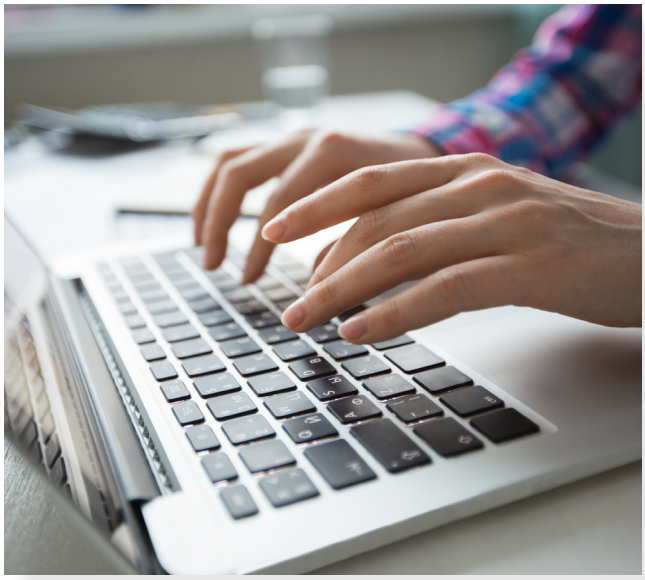
Decision making process for handling SBOM resultsW

APPROVE/REJECT

Let's start with the more clear-cut decisions (in the top left of the diagram above.) If the SBOM and vulnerability report indicate an unacceptable number of high severity vulnerabilities and the risk is just too high to use as-is, then the product should be rejected. Similarly, if the product indicates only minor concerns, then the product can be accepted, with the usual caution that should accompany new products.

CONDITIONALLY APPROVE

In the case where products do have some security issues (see top right above) but the risks outweigh the business needs for the software, the product can be conditionally approved. In these cases, the corporate information security team can implement compensating security controls for the deployment of the product by monitoring potential threat activity targeting known vulnerabilities. Additionally, working with the vendor to remediate the risk is essential as they may be unaware of these vulnerabilities. Disclosure and cooperation are key.



CONDITIONALLY REJECT

If the software product is business-critical but the security risk is just too high (see bottom half above), the product can be conditionally rejected. In such cases, the decision to proceed with deployment will depend on just how critical the software is to the business. In the case where security importance is too high, then you can insist the issues are fixed before deployment, (bottom right above). Or, are you willing to wait for a new version of the software that addresses the most critical security issues before accepting it. Having tools, such as CodeSentry, keeps vendors honest as you can independently verify the issues are indeed resolved.

In the extreme case where the software product is critical to the business and required for daily operations, then it might make sense to negotiate financial, legal and liability terms with its use with the vendor (bottom left above). For example, financial compensation for a data breach, a significant amount which has both a business and reputation impact on you as the customer. Although relatively new territory, this is the reality that software vendors need to adjust to as customers start to improve their software supply chain security. The days of blindly trusting software vendors are over as the risk is far too great.

SUMMARY

The annual spend on enterprise software is hundreds of billions. Yet, enterprises are spending a pittance on securing their software supply chain versus this investment. Most, if not all, commercial products use some sort of open source in their composition and 85% of these applications were found to have critical vulnerabilities according to recent research. Clearly more has to be done to improve the security the software supply chain and ensure a strong enterprise security posture.

Traditionally, software vendors were trusted to deliver secure products. However, those days of blind trust are gone. Despite the best intentions of software vendors, too many security vulnerabilities are lurking in COTS software. The key artifact needed with software procurement is the software bill of materials (SBOM).

As SBOMs gain traction in the marketplace the next logical step is to use the results to improve your software supply chain from the procurement process to what is already deployed today. Using the leverage that a tSBOM provides, it's now possible to hold software vendors accountable. Evaluating software in terms of a risk quadrant is recommended with various accept and reject scenarios. Based on business needs and risk profile, you can make more informed decisions on software supply chain security to proactively reduce risk and eliminate threats in the software that runs your business.

For more information or to request a CodeSentry demo, contact Code Secure or visit

www.CodeSecure.com

Email: sales@codesecure.com