TAG CYBER

IMPLEMENTING SOFTWARE ATTACK SURFACE MANAGEMENT USING THE REZILION DYNAMIC SOFTWARE BILL OF MATERIALS

EDWARD AMOROSO, TAG CYBER

TAG CYBER

IMPLEMENTING SOFTWARE ATTACK SURFACE MANAGEMENT USING THE REZILION DYNAMIC SOFTWARE BILL OF MATERIALS

EDWARD AMOROSO

Software Attack Surface Management (SASM) offers an effective means for securing software throughout an organization's software ecosystem and Software Development Life Cycle (SDLC). The Rezilion Dynamic Software Bill of Materials (SBOM) is shown to implement SASM for practical enterprise environments.

INTRODUCTION

Cybersecurity involves the protection of valued assets from malicious threats. This can be done in a preventive manner with the goal of avoiding attacks, or from a detection and reactive perspective with the objective to deal with attacks once they have begun. These tasks are performed to reduce cyber risk for devices, systems, services, and infrastructure – and the controls are generally managed by an IT security team.

In the context of DevOps, the primary asset to be protected is the software being written. This is often done by adding security controls to the tools, systems, processes, and infrastructure surrounding the software. The primary protection objective is to identify, validate, prioritize, and remediate vulnerabilities while software is being created, tested, released, and managed through the DevOps lifecycle.

To provide guidance on how to implement automated and integrated security controls in DevOps, the concept of *SASM* is introduced here. It draws some parallels with comparable Attack Surface Management (ASM) methods used to secure an enterprise network. The details of SASM and its use of a dynamic SBOM to guide security are illustrated and shown to be effectively implemented by the Rezilion[1] platform.

WHAT IS SOFTWARE ATTACK SURFACE MANAGEMENT (SASM)?

As a result of digital transformation, software has become the number one attack surface. Recent events such as the Log4j (CVE-2021-44228)[2] and SolarWinds[3] breaches provide a direct testament to the importance of protecting this growing attack surface. Explosive growth of the software attack surface and the growing number of attack vectors precipitate the need for tools that can manage and protect the software ecosystem under a unified software attack surface management platform.

The software attack surface is defined as the entire software ecosystem of an enterprise across their entire technology stack such as cloud workloads, hosts, and applications. In order to protect from threats and vulnerabilities, the software attack surface needs to be continuously managed throughout the SDLC from development to production. This practice of driving full stack and full cycle continuous software security is referred to as SASM.

The goal for SASM is to understand the attack surface, identify, prioritize, and remediate vulnerabilities, and to ensure continuous coverage via automation. The dynamic nature of the software attack surface makes it very difficult to manage. One challenge of the software attack surface is the difficulty to manage due to lack of a perimeter for software, a constantly changing feature set, and the growing volume of software.

WHAT IS A DYNAMIC SBOM?

The dynamic SBOM serves as a powerful starting point for discovery and understanding of a software attack surface. A dynamic SBOM is comprised of an inventory of software components such as packages, libraries, files, containers, and images present within the environment that were used to create and manage software, thus providing insight into potential vulnerabilities.

The dynamic SBOM also provides contextual information that offers enriched insight into the potential exploitability of a discovered vulnerability in a specific environment, underlining the fact that the mere presence of a vulnerability does not make it exploitable. This contextualization is built in and is also offered as an independent artifact termed as VEX (Vulnerability Exploitability eXchange) by NIST. Because the dynamic SBOM is continuous and updated in real time as the code makes its passage through the DevOps lifecycle, this insight becomes valuable to developers and security engineers during all phases of the DevOps lifecycle.

One key concept of the dynamic SBOM is provenance, which involves an historical recording of the component origins for software. Thus, where an SBOM provides insights into software components, provenance provides insights into where the software came from, including what changes were made to the code and who made these changes. Provenance offers insights into potential vulnerabilities that might have emerged during the software supply chain.

Dynamic SBOMs are also searchable, a very valuable feature when combined with the continuous and runtime ability of the dynamic SBOM. Using this feature, organizations can search for the presence of specific vulnerabilities such as Log4j and find out if they are loaded to memory and exploitable, or whether it has been patched or remediation action has to be taken.

A final concept of interest is interdependency, which offers insights into how software components communicate, connect, and operate during runtime. Most software tools address the challenge of viewing interdependency by creating a map. Because maps are just graphs, the variables can be anything of interest to the software or security engineer, including connecting software components to their key sources of data. This helps with vulnerability prioritization.

DYNAMIC SBOM AND SASM

Using the dynamic SBOM approach properly for DevOps requires attention to a variety of different lifecycle activities, each of which contributes to reduction of software attack surface risk. These SASM activities include the following support functions:

- **Continuous Integration** – Use of DSBOMs during the integration phase involves the creation and planning phases of DevOps, with attention to the generation and verification of SBOMs, application of SBOM policies, and decisions regarding management and workflows related to the SBOMs.
- **Continuous Deployment** – Dynamic SBOM usage during continuous deployment supports security decision-making regarding SBOMs (e.g., how verification is handled), as well as engagement with risk assessments and decision-making regarding required patches.
- **Continuous Delivery** – Using dynamic SBOM during continuous delivery supports on-going and continuous monitoring for bugs and vulnerabilities, including the need to rework SBOM metadata based on the results of analyses.
- **Continuous Operation** – Dynamic SBOM usage during continuous delivery supports the build process including use of SBOM data to determine vulnerabilities that might be included in deployed software.

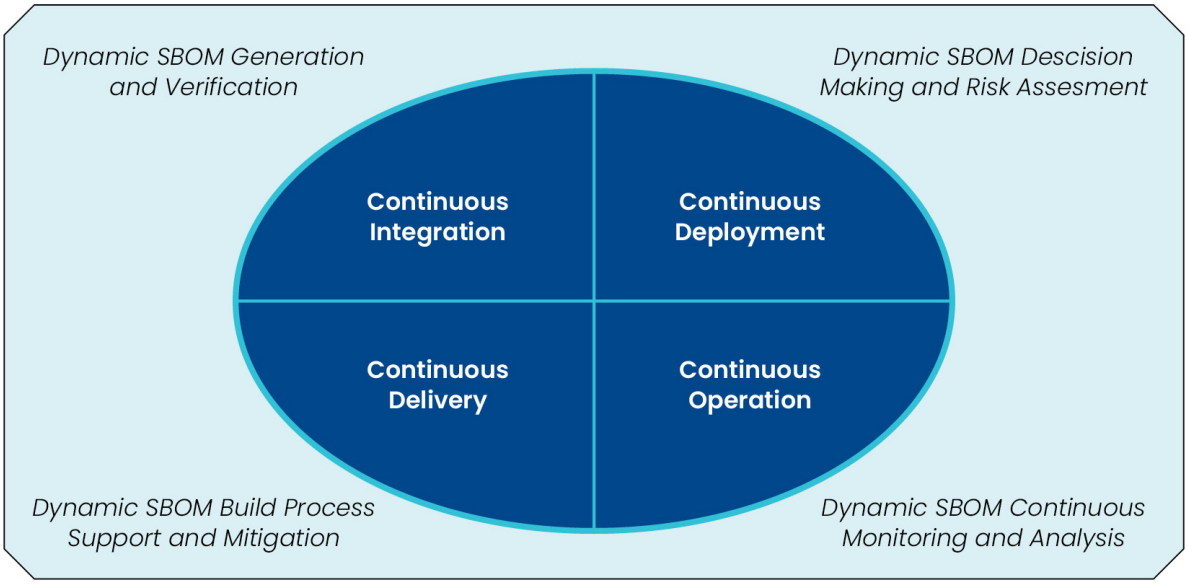

Figure 1. Dynamic SBOM driving security through SDLC

Modern development and product security teams now recognize the importance of these cybersecurity objectives for software during build, release, and operations. In fact, cyber risk management is no longer an option in a typical SDLC but is rather a requirement. To that end, commercial platforms have emerged to support this security goal – and in the next section, we introduce the Rezilion solution and how it implements SASM.

REZILION PLATFORM

Co-founded by former executives from CyActive[4] and headquartered between the US and Israel, cybersecurity start-up Rezilion is focused on empowering organizations to optimize their DevSecOps process through automated security. The goal is to reduce manual dependencies to secure the development, release, and operational aspects of the modern software development life cycle. The Rezilion approach to SASM using their Dynamic SBOM is outlined below.

Consistent with the discussion above, the Rezilion platform drives a SASM process for DevOps by using a dynamic SBOM approach. The platform collects relevant data about the software environment using static and dynamic miners. Runtime and memory data from hosts, containers, and applications are then analyzed for file paths, command-line arguments, hashes, and numerical representations of memory components.

The platform then reverse engineers the software using collected data to map its components, establish vulnerability context, generate provenance, and create the dynamic SBOM. Insights are thus provided to the software and security engineering teams into runtime execution profiles and code interdependencies that help reduce vulnerability backlogs, prioritize what to fix first, and remediate more quickly. Such capabilities would have been especially useful for organizations dealing with the Log4j and SolarWinds incidents cited earlier.

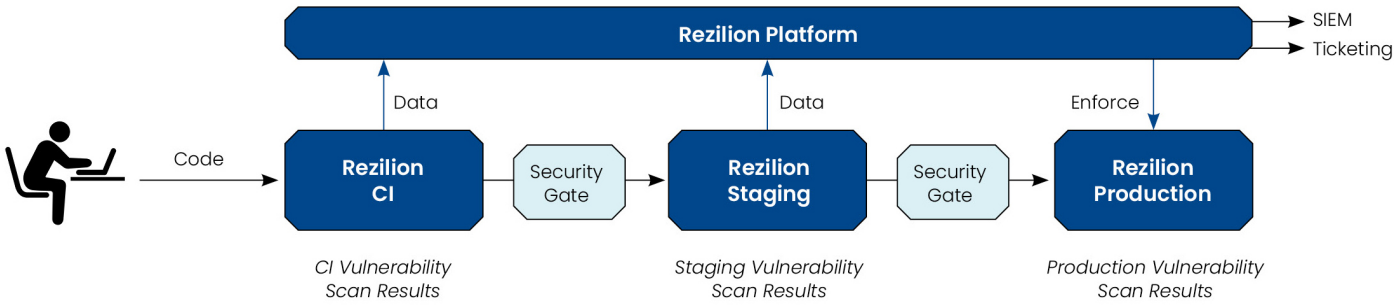

Figure 2. Rezilion Platform Overview

Additional functional SASM advantages include identification of software components, assessment for vulnerabilities, analysis for exploitability (including implementing runtime analysis of what's loaded to memory and what's not), offering automated remediation by aggregating the different vulnerable components, enforcing security policies based on acceptable risk, and detecting drifts and changes at different stages of the SDLC.

Cybersecurity is driven by the Rezilion dynamic SBOM via identification of all software components, mapping of discovered vulnerabilities to provide context, tracking of changes in the software, and maintenance of continuous updates. As any software professional will attest, these capabilities provide essential risk management benefits and are rapidly becoming a priority in any software process environment.

Readers who wish more detailed technical information on the use of Rezilion's Dynamic SBOM to address vulnerabilities like Log4Shell are advised to review this excellent blog by Rezilion[5].

NEXT STEPS: ACTION PLAN

Software engineering and security teams are advised to develop a plan to determine how best to leverage dynamic SBOMs and SASM solutions to reduce cyber risks to their code during DevOps. While the context will vary between different environments, most DevOps teams will benefit by following the steps listed below, which are designed to guide the proper selection and implementation of commercial SASM platforms.

Step 1: Inventory of DevOps Security Controls
The software engineering and security teams are advised to begin with an assessment and inventory of how DevOps security is being handled today. Emphasis should be placed on identifying gaps or places where a control is implemented using a manual process.

Step 2: Vendor Assessment and Review
The teams are next advised to perform a review of available commercial SASM platforms, including Rezilion. TAG Cyber consultants are always available to assist with the tedious process of identifying suitable vendors and comparing strengths and weaknesses.

Step 3: Implementation Planning
The implementation planning will differ from one DevOps team to another, but a phased introduction is always recommended, especially for larger, more complex organizations. The goal, of course, is to deploy SASM quickly to begin providing greater assurance for software in development and use.

---

[1] *https://www.rezilion.com/*
[2] *https://www.wired.com/story/log4j-flaw-hacking-internet/*
[3] *https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack*
[4] *CyActive focused on predictive malware detection and was acquired by PayPal in 2015 (see https://techcrunch.com/2015/03/10/paypal-confirms-acquisition-of-cyactive-plans-to-open-new-security-hub-in-israel/).*
[5] *https://www.rezilion.com/blog/from-0-to-log4j-vulnerability-management-3-easy-steps-in-3-minutes/*

ABOUT TAG CYBER

TAG Cyber is a trusted cybersecurity research analyst firm, providing unbiased industry insights and recommendations to security solution providers and Fortune 100 enterprises. Founded in 2016 by Dr. Edward Amoroso, former SVP/CSO of AT&T, the company bucks the trend of pay-for-play research by offering in-depth Research as a Service (RaaS), market analysis, consulting and personalized content based on hundreds of engagements with clients and nonclients alike—all from a former practitioner perspective.