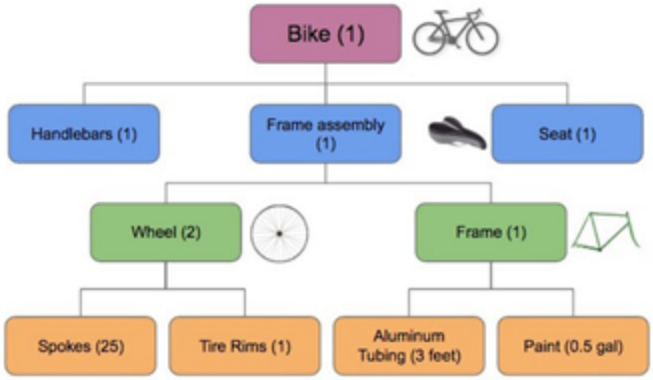
COME CREARE E GESTIRE UNA "SBOM"

I fornitori di software spesso creano prodotti assemblando software open source o commerciali, tool e librerie. La cosiddetta SBOM (Software Bill of Material) è la lista di questi componenti fondamentali per la realizzazione e l'utilizzo del software in forma sicura e razionale

*A cura di Massimo Nannini**

Avete mai sentito parlare di distinta base del software (SBOM)? Sicuramente avrete sicuramente sentito parlare della più nota BOM (Bill of Material), come elemento essenziale per la produzione di un bene. La distinta base la struttura che descrivere nei minimi particolari la composizione dell'oggetto da produrre mettendo in evidenza le relazioni tra i vari componenti, le numerosità dell'utilizzo dei sotto-componenti, le dipendenze gerarchiche, gli indici di revisione, la tipologia di prodotto ecc.



Esempio di SBOM multilivello (OpenBOM)

Che cosa è una SBOM (Software Bill of Material)?

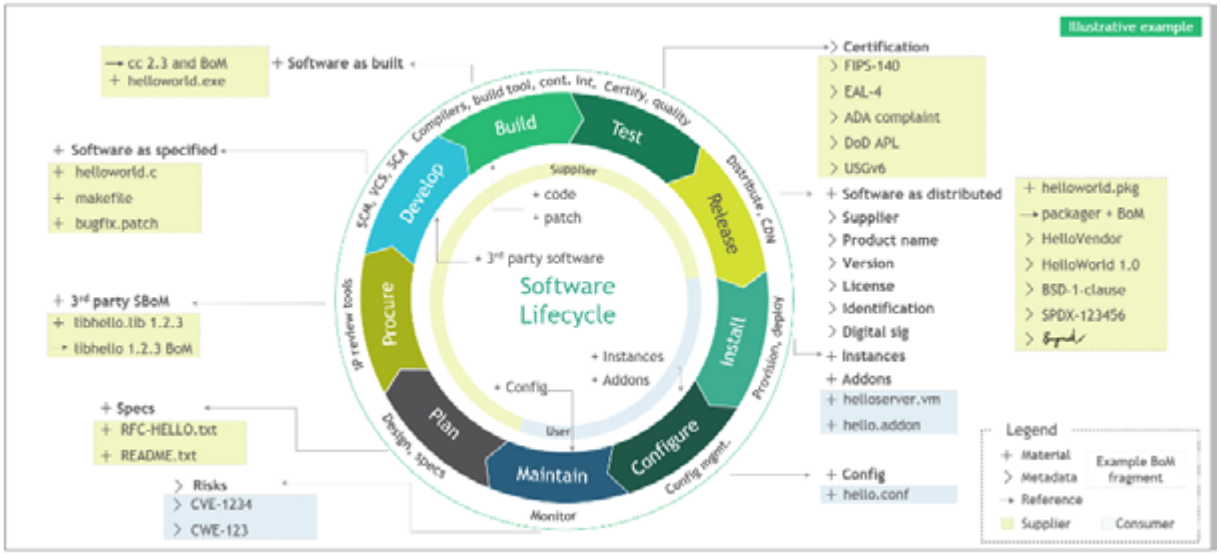
La distinta base del software rappresenta l'"inventario" di tutti i componenti costruttivi e delle dipendenze software coinvolte nello sviluppo e nella consegna di una applicazione diventando così un elemento sempre più comune ed importante nel contesto del ciclo di vita dello sviluppo software (SDLC) e dei processi DevOps.

La crescente complessità delle moderne applicazioni rende questo strumento un punto focale per tenere sotto controllo tutte le parti utilizzate. Tali componenti e dipendenze possono includere progetti software open source, codice proprietario, API, framework e librerie.

Tutti elementi che compongono il prodotto software e che fanno parte della catena di fornitura del software stesso, fungendo da fonti di approvvigionamento per l'abilitazione di applicazioni e servizi.

La figura sotto riportata illustra come una SBOM possa essere costruita attraverso le varie fasi di un processo SLDC (Software Development Life Cycle).

L'obiettivo di una SBOM è dunque fornire l'elenco accurato questi componenti, fornendo agli utenti del software visibilità su ciò che è incluso in un prodotto software e consentendo



Esempio illustrativo di SDLC e SBOM (NIST, National Institute of Standards and Technology)

loro di evitare componenti che possono essere dannosi per motivi di sicurezza o legali.

Perché SBOM è importante per la sicurezza?

Tradizionalmente, le organizzazioni sviluppavano applicazioni internamente utilizzando software proprietario senza avvalersi se non in minima parte di software di terze parti. Questa metodologia ha consentito agli sviluppatori e ai professionisti della sicurezza di tenere sotto controllo l'intera base di codice.

Tuttavia, questo modello non è più in grado di soddisfare le odierne richieste di time-to-market nella produzione del software e dunque è sempre più frequente l'utilizzo di software open source all'interno delle applicazioni.

Il software open source facilita cicli rapidi di sviluppo e rilascio, consente alle organizzazioni di incorporare componenti già pronti nel proprio stack di applicazioni in modo da poter rilasciare rapidamente il prodotto, ma come è ovvio rende più difficile tenere sotto controllo l'intero codice.

Utilizzare la SBOM consente alle organizzazioni di identificare e tenere traccia di tutti i componenti di terze parti, in particolare i componenti open source, e di rispettare i requisiti di licenza. Aiuta inoltre a garantire che l'organizzazione non esegua componenti open vulnerabili tenendo traccia degli aggiornamenti e delle patch critiche garantendo così la sicurezza e la conformità del prodotto.

Tre formati standard

La National Telecommunications and Information Administration (NTIA) suggerisce tre formati per la generazione dell'elenco di inventario SBOM che identifica le entità software e i relativi metadati associati:

- Software Package Data Exchange (SPDX) è uno standard aperto per la creazione di un elenco di inventario SBOM contenente tutti i componenti software, le licenze dei componenti, i diritti d'autore e i riferimenti di sicurezza.
- OWASP CycloneDX è uno standard SBOM leggero per la creazione di un inventario completo di componenti software proprietari e di terze parti per l'analisi dei rischi. CycloneDX può documentare tipi di componenti tra cui applicazioni, contenitori, librerie, file, firmware, framework e sistemi operativi.
- Standard per l'identificazione del software (SWID) è stato sviluppato dall'Organizzazione internazionale per la standardizzazione (ISO) e dalla Commissione elettrotecnica internazionale (IEC). SWID è un file XML contenente un elenco di componenti software e relative licenze, stati delle patch e pacchetti di installazione.

Cosa contiene una SBOM?

Oltre ai suddetti formati la National Telecommunications and Information Administration (NTIA) ha definito le informazioni minime che devono essere incluse in una SBOM:

- Nome dell'autore: in genere l'organizzazione che sviluppa il software.
- Nome fornitore: il nome del fornitore del software, inclusi gli alias (nomi alternativi). Il fornitore e l'autore possono essere diversi se un fornitore sta creando una SBOM per conto del fornitore.
- Nome componente: il nome e gli eventuali alias del componente software.
- Stringa di versione: il formato delle informazioni sulla versione è in formato libero,
- ma dovrebbe seguire l'uso comune del settore.
- Component Hash: il modo migliore per identificare un componente software è utilizzare un hash crittografico che funge da identificatore univoco.
- Identificatore univoco: oltre all'hash, ogni componente deve avere un numero ID che lo identifichi all'interno della SBOM.
- Relazione: definisce la relazione tra il componente e il pacchetto. Nella maggior parte dei casi, la relazione è "inclusa" nel senso che un determinato componente è incluso in un determinato pacchetto.

Una SBOM può includere inoltre informazioni aggiuntive come punteggi di sicurezza, vulnerabilità comuni e codici di esposizione (CVE) di vulnerabilità note nei componenti software e la loro gravità.

Struttura SBOM

Perché usare la SBOM

- Evita il riutilizzo di componenti vulnerabili nei progetti software
- Aiuta a scoprire parti vulnerabili nelle applicazioni correnti
- Gestisce meglio il rischio della catena di fornitura del software attraverso la conoscenza di tutti i componenti (e delle loro dipendenze) utilizzati nelle applicazioni software
- Aiuta le organizzazioni a rispettare meglio le varie normative sulla protezione dei dati fornendo un elenco di tutti i componenti dell'applicazione e le relative caratteristiche
- Aiuta le organizzazioni a selezionare i fornitori

di software preferiti che forniscono le SBOM delle loro applicazioni
- Aiuta le organizzazioni a mantenere un inventario di tutte le applicazioni software nel loro ambiente IT
- Risparmia tempo e risorse delle organizzazioni rilevando le parti vulnerabili nelle prime fasi di progettazione del ciclo di vita dello sviluppo del software (SDLC)
- Richiama la necessaria attenzione sui rischi per la sicurezza associati alla catena di fornitura del software

Keywords: BOM, SBOM, SLDC (Software Development Life Cycle), SPDX (Software Package Data Exchange), OWASP CycloneDX, DevOps, API, Hash, Cybersecurity, Framework, NIST (National Institute of Standards and Technology)

Massimo Nannini*
Ingegnere elettronico e consulente di impresa
info@gemaxconsulting.it

HOW TO CREATE AND MANAGE A "SBOM"

Software vendors often create products by assembling open source or commercial software, tools and libraries. The so-called SBOM (Software Bill of Material) is the list of these components that are essential for the implementation and use of software in a safe and rational form.

Massimo Nannini (*)

Have you ever heard of software bill of materials (SBOM)? Surely you have heard of the better known BOM (Bill of Material), as an essential element in the production of an asset. The BOM the structure that describes in detail the composition of the object to be produced by highlighting the relationships between the various components, the numerosities of the use of sub-components, hierarchical dependencies, revision indices, product type, etc.

What is a SBOM (Software Bill of Material)?
The software bill of materials represents the "inventory" of all building components and software dependencies involved in the development and delivery of an application, thus becoming an increasingly common and important element in the context of the software development life cycle (SDLC) and DevOps processes.
The increasing complexity of modern applications makes this tool a focal point for keeping track of all the parts used. These components and dependencies may include open source software projects, proprietary code, APIs, frameworks, and libraries.
These are all elements that make up the software product and are part of the software supply chain, serving as sources of supply for enabling applications and services.
The figure below illustrates how an SBOM can be built through the various stages of a SLDC (Software Development Life Cycle) process.
The goal of an SBOM is thus to provide the accurate list of these components, providing software users with visibility into what is included in a software product and enabling them to avoid components that may be harmful for security or legal reasons.

Why is SBOM important for security?

Traditionally, organizations developed applications internally using proprietary software with little or no use of third-party software. This methodology allowed developers and security professionals to keep the entire code base under control. However, this model is no longer able to meet today's time-to-market demands in software production, and therefore the use of open source software within applications is increasingly common.

Open source software facilitates rapid development and release cycles, allows organizations to incorporate off-the-shelf components into their application stack so they can quickly release the product, but as is obvious it makes it more difficult to keep track of the entire code.

Using the SBOM allows organizations to identify and track all third-party components, especially open source components, and to comply with licensing requirements. It also helps ensure that the organization does not run vulnerable open components by keeping track of critical updates and patches thus ensuring product security and compliance.

Three standard formats

The National Telecommunications and Information Administration (NTIA) suggests three formats for generating the SBOM inventory list that identifies software entities and their associated metadata:

- Software Package Data Exchange (SPDX) is an open standard for creating an SBOM inventory list containing all software components, component licenses, copyrights, and security references.
- OWASP CycloneDX is a lightweight SBOM standard for creating a complete inventory of proprietary and third-party software components for risk analysis. CycloneDX can document component types including applications, containers, libraries, files, firmware, frameworks, and operating systems.
- Standard for Software Identification (SWID) was developed by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). SWID is an XML file containing a list of software components and their licenses, patch states, and packages for
- installation.

What does a SBOM contain?

In addition to the above formats, the National Telecommunications and Information Administration (NTIA) has defined the minimum information that must be included in an SBOM:

- Author name: usually the organization developing the software.
- Vendor name: the name of the software vendor, including aliases (alternate names). The supplier and author may be different if a supplier is creating an SBOM on behalf of the supplier.
- Component name: the name and any aliases of the software component.
- Version string: the format of the version information is in free format, but should follow common industry usage.
- Component Hash: the best way to identify a software component is to use a cryptographic hash that serves as a unique identifier.
- Unique identifier: in addition to the hash, each component must have an ID number that identifies it within the SBOM.
- Relationship: defines the relationship between the component and the packet. In most cases, the relationship is "included" in the sense that a given component is included in a given package.

A SBOM may also include additional information such as security scores, common vulnerabilities and exposure codes (CVEs) of known vulnerabilities in software components and their severity.

Why use the SBOM

- Prevents reuse of vulnerable components in software projects
- Helps discover vulnerable parts in current applications
- Better manages software supply chain risk through knowledge of all components (and their dependencies) used in software applications
- Helps organizations better comply with various data protection regulations by providing a list of all application components and their characteristics
- Helps organizations select preferred software vendors that provide SBOMs of their applications
- Helps organizations maintain an inventory of all software applications in their IT environment

- Saves organizations' time and resources by detecting vulnerable parts early in the design phases of the software development life cycle (SDLC)
- Draws necessary attention to security risks associated with the software supply chain

Keywords: BOM, SBOM, SLDC (Software Development Life Cycle), SPDX (Software Package Data Exchange), OWASP CycloneDX, DevOps, API, Hash, Cybersecurity, Framework, NIST (National Institute of Standards and Technology)

Consigli di lettura

Mario Gargantini - Carlo Marchisio

Automation Story

Le tecnologie, gli uomini, le imprese dell'automazione

Codice: AUT
ISBN: 978-88-97323-36-5
Prezzo: 20,00 €
Edizione: Ristampa 2022
Formato: 17 x 24
Pagine: 208

Il volume racconta l'evoluzione delle tecniche e dei sistemi per il controllo dei processi produttivi: dai primi tentativi dell'antichità, ai regolatori per le macchine a vapore della prima rivoluzione industriale alla strumentazione che ha dominato l'era dell'elettricità; fino i PLC, ai DCS, alla meccatronica e all'incontro con l'Information Technology. Macchine e strumenti che hanno reso possibile ottimizzare la produzione nei settori più diversi: dall'automotive al packaging, da food all'energia. E alla base delle macchine, gli uomini: da inventori come Watt o Tesla, a imprenditori come Siemens o Bradle o Bosch scienziati come Wiener, padre della cibernetica.

Editoriale Delfino
Tel. 02 9578.4238
info@editorialedelfino.it