

Software Bill of Materials from a Software Configuration Management Perspective

Lars Bendix, Lund University
Andreas Göransson, QCM
sneSCM.org

https://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Research/SBoM/

Agenda

- SBoM motivation
- SBoM history
- SBoM Use Case categories
- General SBoM considerations

- SBoM from an SCM perspective
- SBoM from a Developer perspective
- Shift left Cyber Security

- Q&A

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024




SBoM motivation



- Should have been an *internal* requirement since the 1980's
- **Developers** should be heavy *consumers* of SBoMs
- SBoMs needed for all systems in environment (know what you have)
- If you know what you have, then change control (and cyber security) is easier
- **Developers** are the optimal *producers* of SBoMs
- How you can *use/exploit – and (help) create* - SBoM - the full story

These are examples of SBoMs



 Version 5.13.7 (15481) Copyright ©2012-2023 Zoom Video Communications, Inc. All rights reserved.	Update Available New version 5.13.10.16307 is available. You have 5.13.7 (15481). Release notes of 5.13.10 (16307) Changes to existing features -Breakout Rooms 100 enabled for all accounts General features -Activity Center -Zoom Network Connectivity Tool -Additional MSI/PLIST/GPO/MDM options -Allow users to access Network Diagnostics Tool -Set IP address for Zoom Mesh local detection -Set network port for Zoom Mesh nodes -Set network port range for Zoom Mesh parent-child node communication -Allow external participants to connect via Zoom Mesh Meeting/webinar features -On-demand watermark -Custom watermark location
--	---

SBoM anno 2020s

- Executive Order 14028 signed in 2021
- Strengthen the US' ability to respond quickly and efficiently to cybersecurity vulnerabilities
 - Heartbleed, SolarWinds, Colonial Pipeline hack
- US Department of Commerce and NTIA
 - The Minimum Elements For a Software Bill of Materials
 - “List of ingredients” with focus on vulnerability scans



© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

Fast rewind - Back to the future



© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

SBoM anno 1980s - part I



Wayne Babich: “Many times the fastest approach to finding a bug is not analysis of the program itself, but analysis of the *history* of the program – how it was created. The history of the program is called its *derivation*.”

A precise *derivation* of a program or module requires:

- an identification of the *tool* that created it
- an identification of the data that was *input* to the tool
- an identification of the *options* and *arguments* given to the tool
- the *reason* why that particular data, arguments, and options were given to the tool
- the *person* who was responsible for creating the data
- the *date* and *time*

An ounce of derivation is worth a pound of analysis!

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

Back to the future II



© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

SBoM anno 1980s - part II



- Clearmake automatically create SBoMs called Configuration Records
- Re-use build artifacts like object files in a smart and secure way
- Configuration Record contains information about input files, build environment and output files
- Configuration records can be read and used by machines and audited manually by developers

Overview – SBoM use case categories



- Bo-Materials
 - Search for object by UID
- Bo-Process
 - Reuse
 - Debugging
 - Rebuild
 - Build audit
- Bo-Information
 - Licence tracking
 - Export control
 - Legal aspects
 - Test-related matters
 - Information sharing

BoM use case category



Search for an object by UID

Software Composition Analysis (SCA):

- Are features 12, 15 and 19 included in this binary for QA?
- We get a new (binary) patch - is it already in there?
- What is “operating” on our systems (Ops)

SBoM = **BoM**

BoP use case category



Reproducibility - or producibility - is a core concept for software configuration management

For that we need the exact source code - obviously...

But that is not sufficient:

- Escrow development
- Fixing a bug in a 13-year old petro-chemical installation

To the degree of what you consider “identical” we must include the process:

- Tools, versions, options, environments, ...

SBoM = BoM + **BoP**

BoI use case category



Information about the artifacts in the SBoM needed for communication, audits, certifications, ...

Test related matters:

- In case a program or system can affect human safety, test information can be vital to keep in an SBoM
 - Test cases, test results, test environments (HW and SW), ...
- Possible need to provide proof that test cases have been performed in a legal dispute
- Test information can also be used as a “quality stamp”
 - Our software passes these test and therefore complies to regulation X $SBoM = BoM + BoP + BoI$

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

SBoM implementation considerations



- Level of detail of SBoM
- Availability of SBoM
- **Automation - both in creation and using**
- Static and dynamic status of SBoM data
- Keeping the SBoM up to date
- **SBoM for tools and environments too - and Everything**
- SBoM and Microservices
- Unique identifiers

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

Key takeaways



- SBoMs for **consumers**:
 - External consumers: List of ingredients & Vulnerability scan
 - Internal consumers: 100s of use cases, 10+ categories, 3 aspects
- SBoMs for **producers**:
 - “Builders” of a binary are at the origin of (most) SBoM data
 - Automation is at the heart of Developers
- Doesn't it look a little like:
 - CMDB: Configuration Items with data (attributes and relations)
 - Configuration Status Accounting

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

Looking forward



- Those were the results – so far...
- Leaving a lot of interesting questions to dig deeper into
- Let's dive into it.....

© Lars Bendix, Andreas Göransson – sneSCM.org

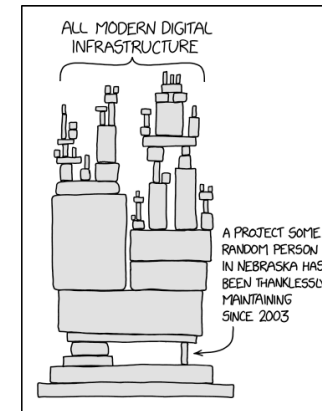
Config Management Camp, Ghent, Belgium, February 5, 2024

Who can benefit from SBoMs?



- Externally (consumers):
 - End users
 - Buyers
 - Applications
 - Libraries
- Internally (producers):
 - Developers
 - Quality Assurance
 - Cyber Security / Supply Chain Security / SCA
 - Software Configuration Management

Missing Configuration Items...



<https://xkcd.com/2347/>

What is an SBoMs?



- A "list of ingredients"++
- A machine-readable inventory of binaries
- Dependencies and relations (traceability)
- Metadata
- Flat – or hierarchical (SBoM of SBoMs)?

How to use (create value from) SBoMs?



- Diff on SBoMs – bug hunting
- Dependencies and relations (traceability)
- Reproducible builds
- Upgrading installations/production
- Risk analysis
- To identify – to notify – and to recall

Who can create an SBoM – and how?



- Those who *build* the application/binary
- At the *time* when they build the application/binary
- *Every* application/binary must have an SBoM

SBoM from an **SCM** point of view – the foundation



- Configuration Items
- Attributes
- Relations
- CMDB

SBoM from an **SCM** point of view – added value



- Configuration Status Accounting: all the questions you never had answers for before
- If we know what we are using (have in our repo), we can ask interesting questions like: Do we need to use 84 versions of Spring? Do we need to have 15 different XML parsers?
- Configuration Change Control
- Configuration Audit
- Baselines – traceability – SBoM
- Adam Thornhill: Code as a Crime Scene (CMDB & CSA)

SBoM from a **Developer** point of view



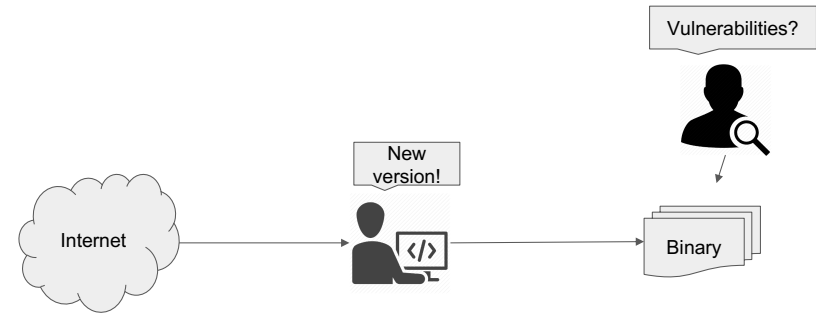
- Developer: what a cool library, I could use that!
- ?When is EOL
- ?What is the maintenance rating
- ?What are the number of dependencies
- ?Will all our code become open source
- ?.....

”Useless” pieces of information?

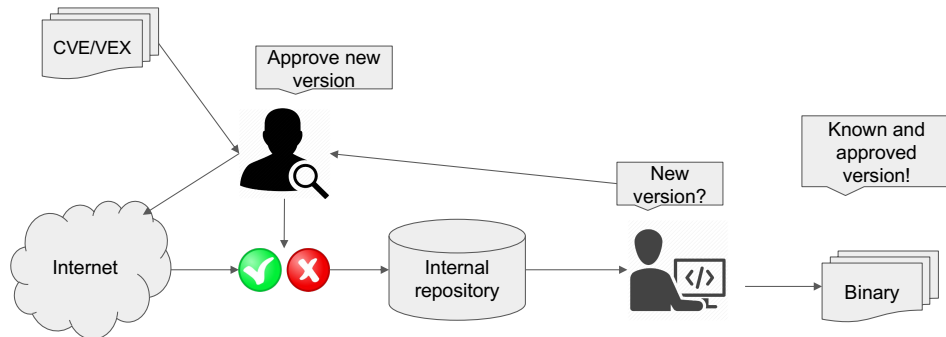


- 80-90% of an application’s code is “external”.
- Six out of seven vulnerabilities come from transitive dependencies.
- Organisations keep fetching known vulnerable versions of components that are already fixed (30% log4j after 18 months).
- CVE / VEX / VDR / GUAC / TACOS / SLSA

Shift left security?



Shift left security!



A couple of interesting quotes?



- Afraid of sharing SBOMs? Couldn’t we just “share” the **libraries** we use (which is probably where the “public” vulnerabilities are)
- “Are we – directly or indirectly – using log4j?”
- Log4j is **not** the normal case – “ordinary” bugs are
- ”Where is this internal library used?”

Key takeaways



- SBoM is much broader than “vulnerability scan”
- an SBoM should be created at the same time as the binary
- automation, automation, automation
- if we “shift left security”, we do “security right”

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024

Re-inventing the wheel/SBoM?



© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024



Thanks - Q&A

https://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Research/SBoM/

© Lars Bendix, Andreas Göransson – sneSCM.org

Config Management Camp, Ghent, Belgium, February 5, 2024