Software Bill of Materials: Visualizing the Unseen

APRIL 5, 2023

Michael Bandor
Senior Software Engineer

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Carnegie Mellon University
Software Engineering Institute

Document Markings

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM23-0344

Agenda

- Software Bill of Materials (SBOM) - Background
- Graph Visualization
- SBOM Data and Graphs: A Closer Look
- SBOM Data – Expectations vs. Reality

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Software Bill of Materials: Visualizing the Unseen

Software Bill of Materials (SBOM) - Background

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Carnegie Mellon University
Software Engineering Institute

What is a Software Bill of Materials (SBOM)

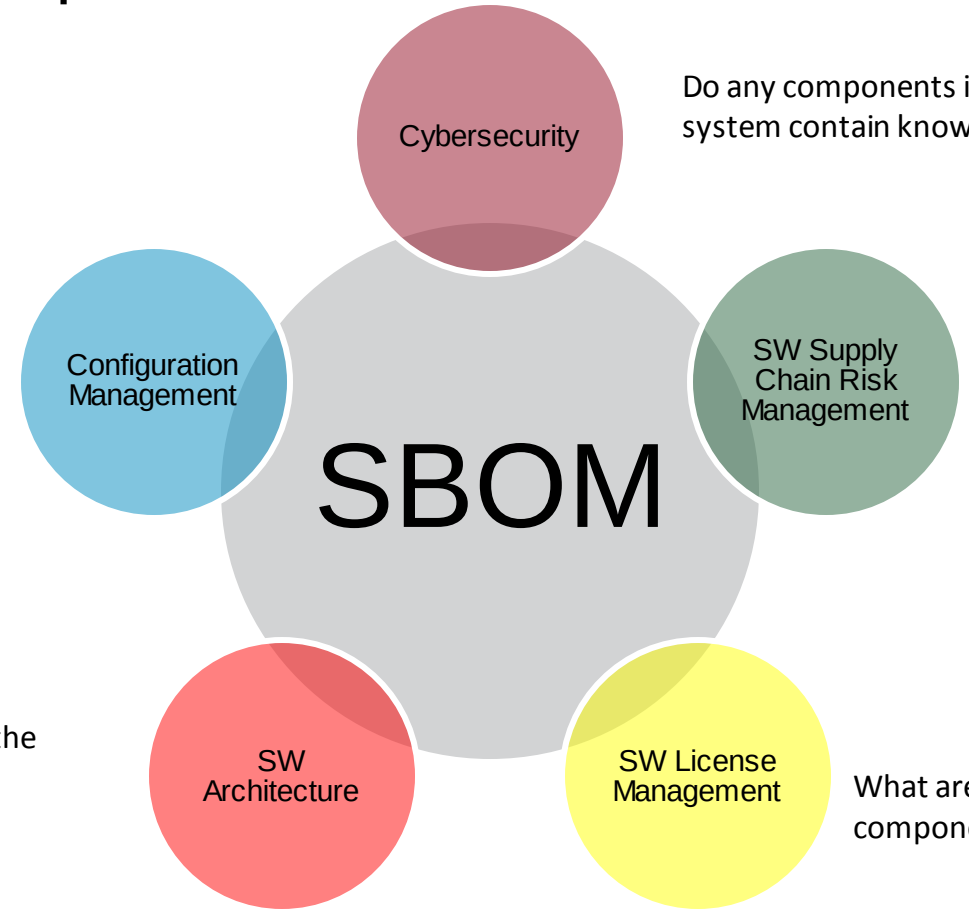
An SBOM is a formal record containing the details and supply chain relationships of various components used in building software. In addition to establishing these minimum elements, this report defines the scope of how to think about minimum elements, describes SBOM use cases for greater transparency in the software supply chain, and lays out options for future evolution.[1]

SBOMs are mandated under a federal directive EO 14028, *Executive Order on Improving the Nation's Cybersecurity*[2]

1 *The Minimum Essential Elements of a Software Bill of Materials*, United States Department of Commerce, July 12, 2021, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf
2 *Executive Order on Improving the Nation's Cybersecurity*, White House, 12 May 2021, https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOM Relationships with Other Areas



Do any components in or used by the system contain known vulnerabilities?

What components are included/needed in my system?

What are the third-party components in the system and the source of the component information?

Where are the components in the system architecture?

What are the licensing/data rights to components in my system?

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOMs at Many Levels

Organizations tend to focus on the product(s) coming through their development pipeline(s)

- What about the tools in the pipeline(s)? Do you know what is there?
- What about the other software used to support the product?
- How do you get complete situational awareness across the entire program?

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

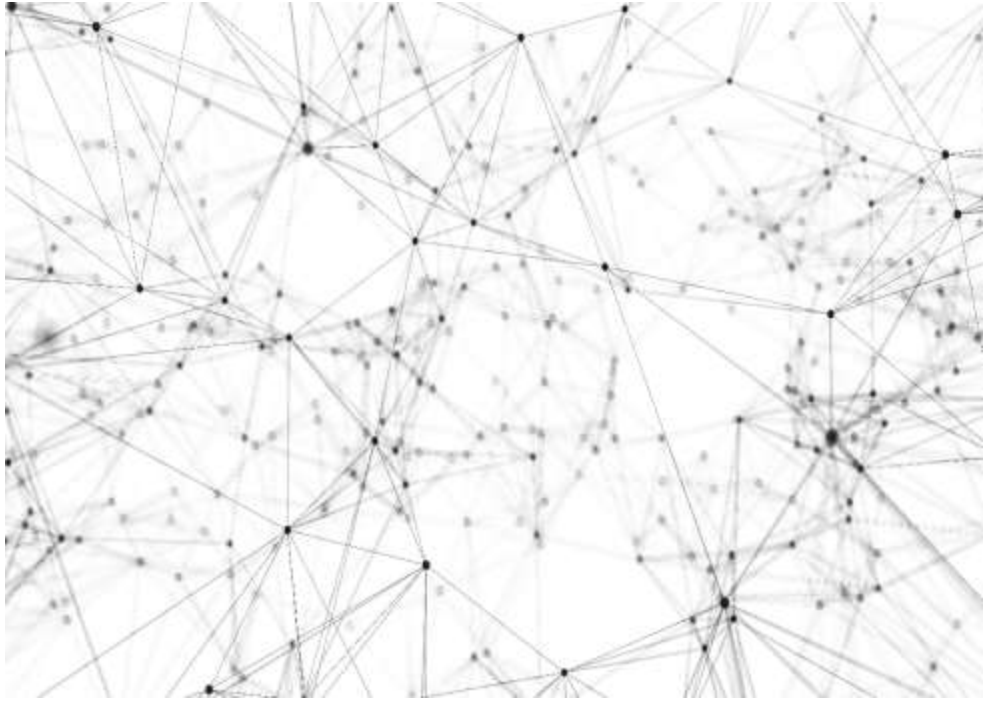
Software Bill of Materials: Visualizing the Unseen

Graph Visualization

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Carnegie Mellon University
Software Engineering Institute

Using Graphs to Visualize the Unknown



Everything is naturally connected, networks of people, transactions, supply chains

> *"Graphs form the foundation of modern data and analytics techniques, with capabilities to enhance and improve user collaboration, Machine Learning models, and explainable Artificial Intelligence." – Gartner, "Top 10 Tech Trends in Data and Analytics," 16 Feb 2021*[1]

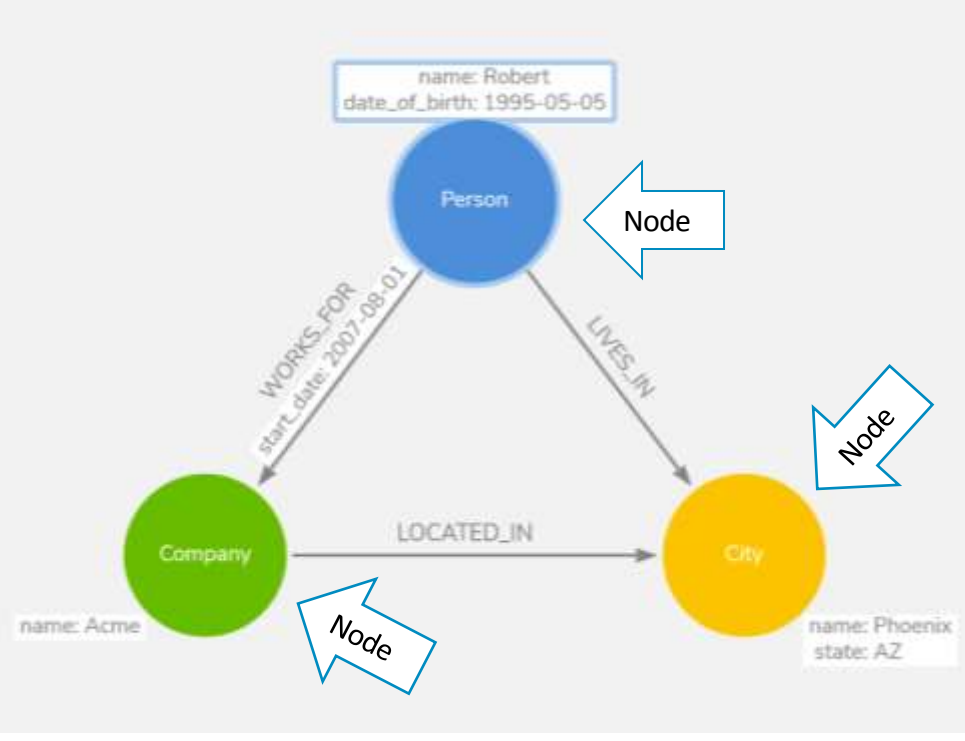
Using graphs encodes relationships that cut across data elements and exposes their critical aspects that would not otherwise be visible

A graph lets the problem be represented through **Nodes** and **Relationships** of the nodes to each other

[1] https://www.gartner.com/smarterwithgartner/gartner-top-10-data-and-analytics-trends-for-2021

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Graph Representation



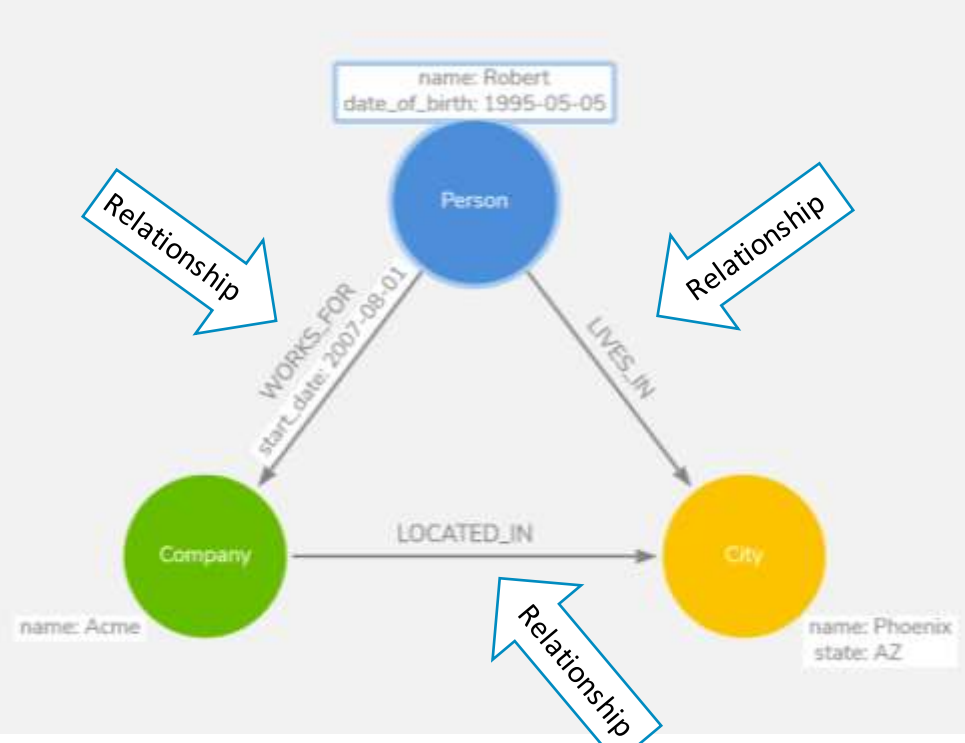
Nodes* are the entities in the graph.

- They represent objects (nouns)
- They can hold any number of attributes (key-value pairs) called properties.
- Nodes can be tagged with labels, representing their different roles in your domain.
- Node labels may also serve to attach metadata (such as index or constraint information) to certain nodes.

* *The Property Graph Model*, https://neo4j.com/developer/graph-database/

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Graph Representation



Relationships* provide directed, named, semantically relevant connections between two node entities (e.g., Employee WORKS_FOR Company).

- Relationships connect nodes and represent actions (verbs)
- A relationship always has a direction, a type, a start node, and an end node.
- Like nodes, relationships can also have properties. In most cases, relationships have quantitative properties, such as weights, costs, distances, ratings, time intervals, or strengths.
- Due to the efficient way relationships are stored, two nodes can share any number or type of relationships without sacrificing performance.
- Although they are stored in a specific direction, relationships can always be navigated efficiently in either direction.

* *The Property Graph Model*, https://neo4j.com/developer/graph-database/

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Uses of Graph Technology

Carnegie Mellon University Software Engineering Institute

- MITRE: Cybersecurity Situational awareness (https://neo4j.com/case-studies/mitre/)
- NASA: Lessons learned and knowledge management (https://neo4j.com/users/nasa/)
- Lyft: Data discovery (https://neo4j.com/case-studies/lyft/)
- Lockheed Martin Space: Lifecycle data and parts management (https://neo4j.com/case-studies/lockheed-martin-space/)
- CAST Software: IT architecture visibility (https://neo4j.com/case-studies/cast-software/)
- US Army: Equipment maintenance tracking (https://neo4j.com/case-studies/us-army/)
- US Dept of Homeland Security (DHS): Information sharing for Enterprise Architects including critical intelligence (https://www.youtube.com/watch?v=aMPm4Zo58E4)

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Software Bill of Materials: Visualizing the Unseen

SBOM Data and Graphs: A Closer Look

Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOMs and Graphs: A Closer Look

A closer review of the guidance reveals the following (highlighting added for emphasis):

> ***Depth**. An SBOM should contain all primary (top level) components, with all their **transitive dependencies** listed. At a minimum, all top-level dependencies must be listed with enough detail to **seek out the transitive dependencies recursively.***
>
> ***Going further into the graph will provide more information.** As organizations begin SBOM, depth beyond the primary components may not be easily available due to existing requirements with subcomponent suppliers. Eventual adoption of SBOM processes will enable access to additional depth through deeper levels of transparency at the subcomponent level. **It should be noted that some use cases require complete or mostly complete graphs, such as the ability to "prove the negative" that a given component is not on an organization's network.***[1]

1 *The Minimum Essential Elements of a Software Bill of Materials*, United States Department of Commerce, July 12, 2021, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOMs and Graphs: A Closer Look

Carnegie Mellon University Software Engineering Institute

Guidance review (continued):

Known Unknowns. ***For instances in which the full dependency graph is not enumerated in the SBOM, the SBOM author must explicitly identify "known unknowns."*** *That is, the dependency data draws a clear distinction between a component that has no further dependencies, and a component for which the presence of dependencies is unknown and incomplete...*

Other Component Relationships. ***The minimum elements of SBOM are connected through a single type of relationship: dependency. That is, X is included in Y. This relationship is implied in the SBOM graph structure…***[1]

1 *The Minimum Essential Elements of a Software Bill of Materials*, United States Department of Commerce, July 12, 2021, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

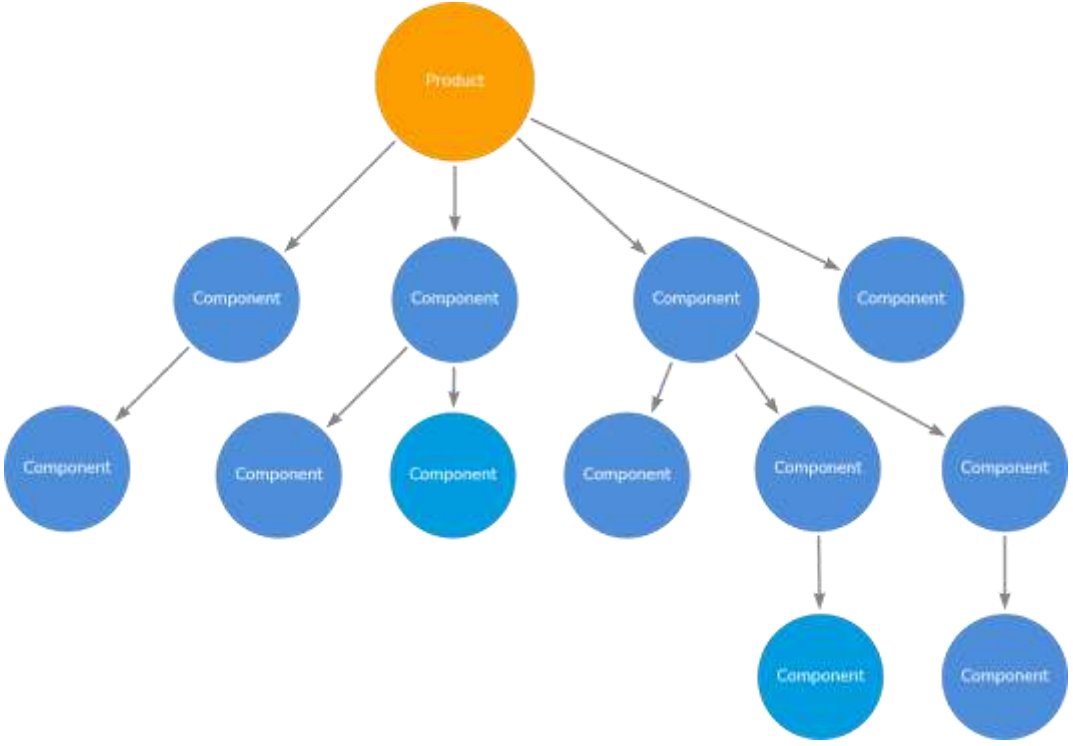
Software Bill of Materials: Visualizing the Unseen

SBOM Dependencies – Expectations vs. Reality

Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOM Dependencies – Expectations vs. Reality



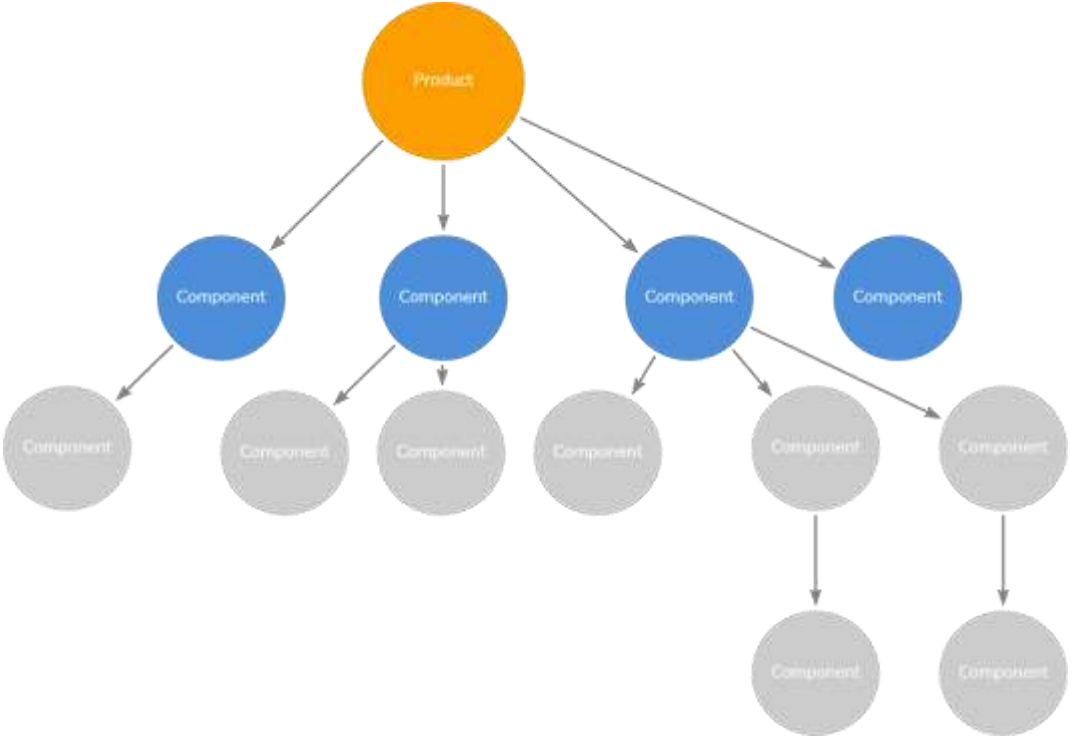
SBOM Dependencies – Organizational Expectations

Many organizations think of their generated SBOM data looks this way

- Assuming the secondary and tertiary dependencies are ingested from their respective suppliers

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOM Dependencies – Expectations vs. Reality



SBOM Dependencies – Organizational Expectations & Missing Data

An organization can currently control what they use

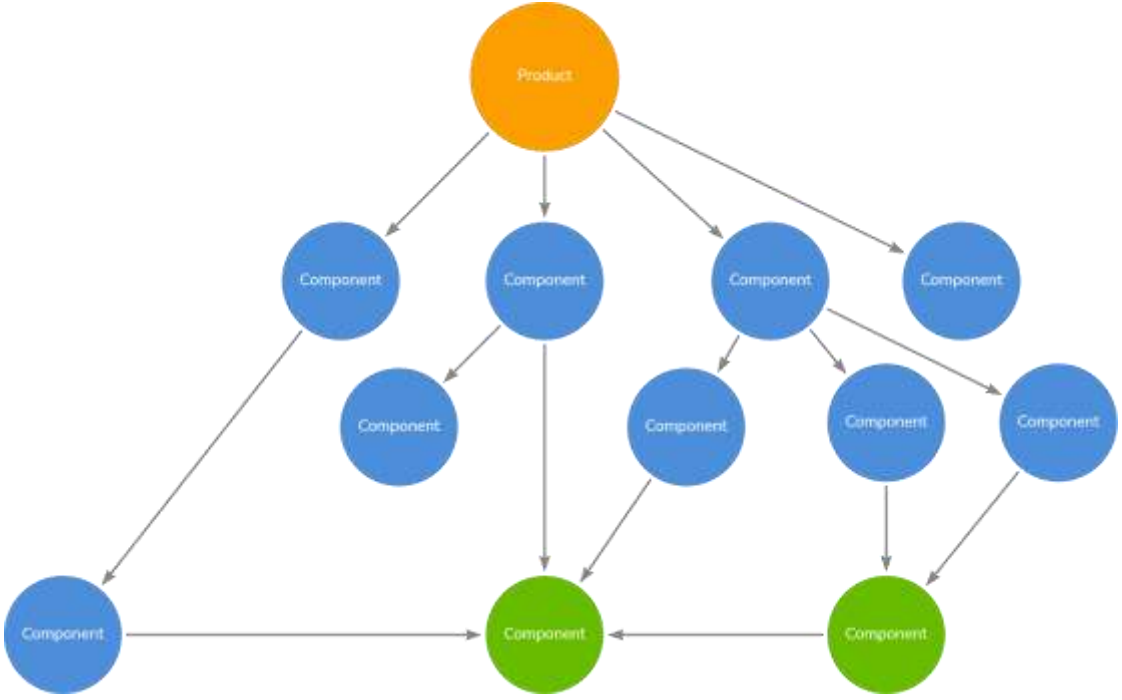
- What are the first level components?
- What are the risks of the unknown components (secondary & tertiary)?

Adding the information to the dependencies is the responsibility of the organization

- Potentially a full-time responsibility for individual(s) to monitor & maintain

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOM Dependencies – Expectations vs. Reality



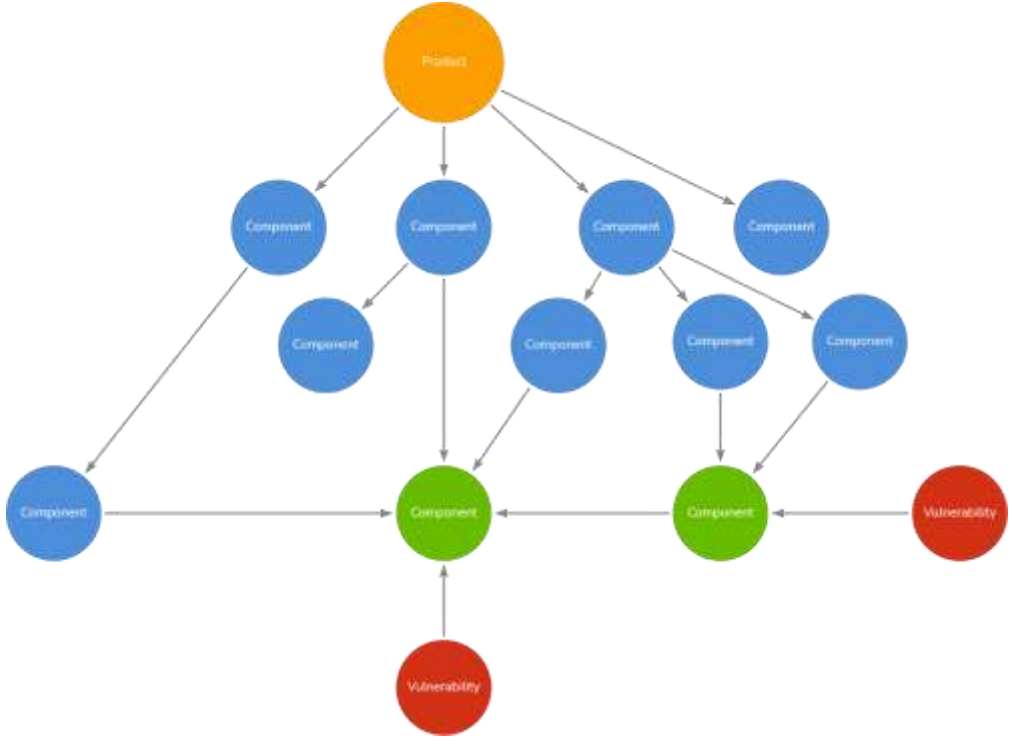
SBOM Dependencies – Reality of the Dependencies

The reality is the component dependencies are not a clean as most organizations think

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

SBOM Dependencies – Expectations vs. Reality



SBOM Dependencies and Vulnerabilities

Adding the vulnerabilities to the data, the system impacts and risks become more apparent!

- Vulnerabilities tend to be managed independently

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Which is Easier to Analyze?

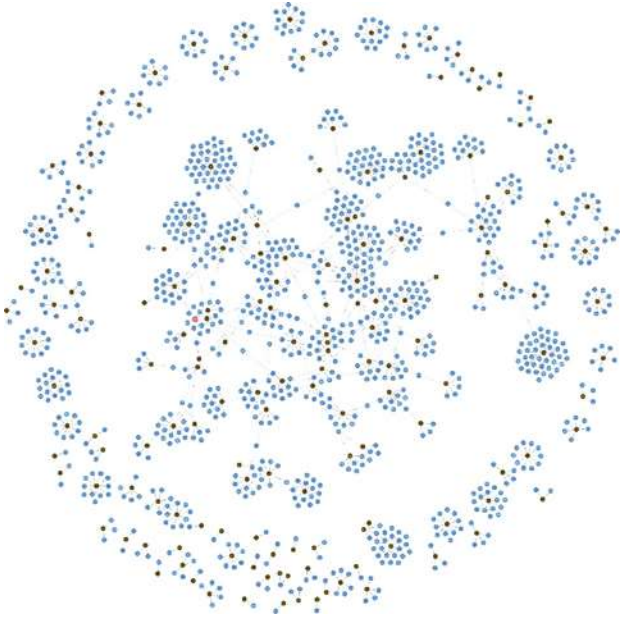
OBS SW Name	OBS SW Version	OBS Vendor	Manufacturer	Risk Category	Is Est. EOS	EOS Date	EOES Date	EOL Date
Administrative Templates (ADMX)	Windows Server 2008 R2 and Windows 7	Microsoft	Microsoft	HIGH	NO	14-Jan-20	14-Jan-20	14-Jan-20
Administrative Templates (ADMX)	Windows Server 2012 and Windows 7	Microsoft	Microsoft	LOW	NO	10-Oct-23	10-Oct-23	10-Oct-23
Administrative Templates (ADMX)	Windows Server 2016 and Windows 10	Microsoft	Microsoft	MEDIUM	NO	30-Oct-18	10-Oct-23	10-Oct-23
Adobe Acrobat Pro	2017	Adobe	Adobe	HIGH	NO	6-Jun-22	6-Jun-22	6-Jun-22
Adobe Flash Player	11.2.202.616 (Linux)	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	21.0.0.213 (Windows)	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	27.0.0.170	Adobe		HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	28.0.0.161 (Linux)	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	28.0.0.161 (Windows with ActiveX)	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	32.0.0.156	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player	32.0.0.223	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player NPAPI	32.0.0.414	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player with ActiveX	32.0.0.223	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player with ActiveX	32.0.0.387	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Flash Player with ActiveX	32.0.0.414	Adobe	Adobe	HIGH	NO	31-Dec-20	31-Dec-20	31-Dec-20
Adobe Reader	11.0.10 (Windows)	Adobe	Adobe	HIGH	NO	15-Oct-17	15-Oct-17	15-Oct-17
Adobe Reader	11.0.19 (Patched from 11.0.10) (Windows)	Adobe	Adobe	HIGH	NO	15-Oct-17	15-Oct-17	15-Oct-17
Adobe Reader	9.5.5-1 (Linux)	Adobe	Adobe	HIGH	NO	26-Jun-13	26-Jun-13	26-Jun-13
Adobe Reader DC	20.013.20074	Adobe	Adobe	LOW	NO	1-Jun-25	1-Jun-25	1-Jun-25
Adobe Reader DC	2018.011.20040	Adobe	Adobe	LOW	NO	13-May-23	13-May-23	13-May-23
Adobe Reader DC	2019.010.20098	Adobe	Adobe	LOW	NO	21-Feb-24	21-Feb-23	21-Feb-24
Adobe Reader DC	2019.012.20036	Adobe	Adobe	LOW	NO	13-Aug-24	13-Aug-24	13-Aug-24
Adobe Reader DC	2020.006.20042	Adobe	Adobe	LOW	NO	1-Jun-25	1-Jun-25	1-Jun-25
Adobe Reader DC	21.005.20048	Adobe	Adobe	LOW	NO	8-Jun-26	8-Jun-26	8-Jun-26
Adobe Reader DC	21.007.20095 (Windows 32-bit)	Adobe	Adobe	LOW	YES	29-Sep-25		
Adobe Reader DC	21.007.20095 (Windows 64-bit)	Adobe		LOW	YES	29-Sep-25		
AF PKI SPO SIPRNet CRL AutoCache Script	4.2	Gov	Gov	LOW	YES	13-Jun-24		
Anaconda Distribution	4.2.0 (Linux 64-bit)	Anaconda	Anaconda	LOW	YES	27-Sep-23		
Anaconda Distribution	4.2.0 (Windows 64-bit)	Anaconda	Anaconda	LOW	YES	27-Sep-23		
ANTLR	3.2	ANTLR	ANTLR	HIGH	NO	20-Sep-16	20-Sep-16	20-Sep-16
Apache ActiveMQ	5.14.3	Apache	Apache	POTENTIAL	YES	22-Dec-21		
Apache ActiveMQ	5.15.12	Apache	Apache	LOW	YES	12-Mar-25		
Apache ActiveMQ	5.8.0	Apache	Apache	POTENTIAL	YES	12-Feb-18		
Apache ActiveMQ-CPP	3.4	Apache	Apache	POTENTIAL	YES	23-Apr-18		
Apache ActiveMQ-CPP	3.9.3	Apache	Apache	POTENTIAL	YES	24-Mar-21		

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

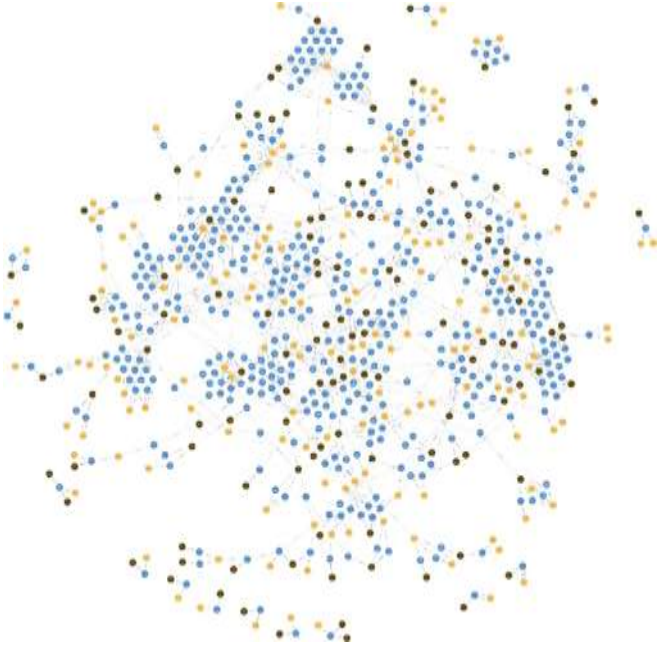
Which is Easier to Analyze?

Carnegie Mellon University Software Engineering Institute

Product End of Support Clusters



Jira Ticket Clusters

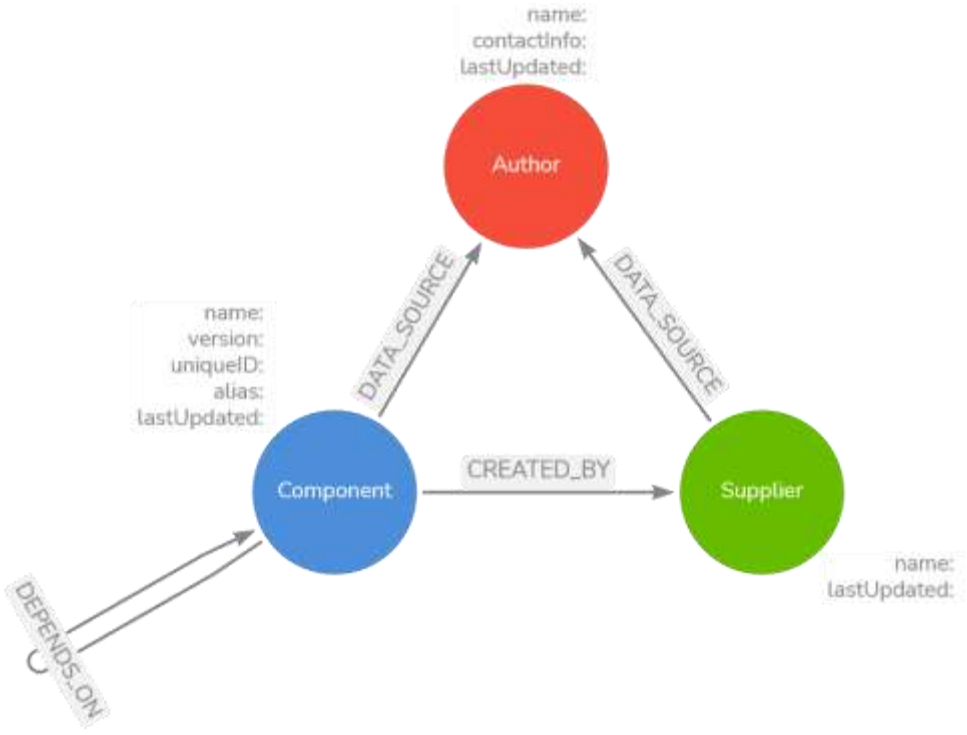


Previously unrealized/unseen clusters of data!

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Further Research – Graphing SBOM Data



Basic SBOM Graph Model

Using data exported from an SBOM tool:

- Ingest the data to create the graph prototype (start with SDPX format)
- Develop the scripts and processes to ingest and refine (update) the data
- Identify vulnerability data and add it to the graph
- Use the basic use cases from the SBOM guidance to prove out the model and prototype

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Questions

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Contact Information

Michael Bandor
Senior Software Engineer

Telephone: +1 412.268.8423
Email: mbandor@sei.cmu.edu

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.