From: Dick Brooks <dick@reliableenergyanalytics.com>
Sent: Tuesday, June 8, 2021 5:03 PM
To: 'SBOM_RFC@ntia.gov' <SBOM_RFC@ntia.gov>
Subject: SBOM Comments of Reliable Energy Analytics LLC (REA) in response to document NTIA-2021-0001

Reliable Energy Analytics LLC (REA) thanks the Department of Commerce (DoC) and the National Telecommunications and Information Administration (NTIA) for the opportunity to provide these comments in response to the June 02, 2021 call for comments regarding Docket # 210527-0117, Software Bill of Materials Elements and Considerations, document identifier NTIA-2021-0001.

REA is a long-time supporter of and participant in the NTIA SBOM initiative and remains actively involved in the NTIA SBOM initiative. REA believes that the software supply chain cannot be adequately secured without an accurate, timely, complete and trustworthy SBOM along with the software package it describes in an immutable bonded trust relationship, that can be verified by a software consumer as part of a comprehensive and effective SCRM cybersecurity software risk assessment.

Currently, most software customers do not require their software vendors to provide SBOM's, when they deliver a software product. This "missing SBOM link" makes it difficult for a software customer to perform an effective SCRM risk assessment on a software package due to the following critical issues:

- No uniform, standard method to identify the legitimate Software Supplier of a software package delivered to software customers
- Many existing software packages do not explicitly identify the legitimate software supplier
- Many software packages delivered to customers do not "list their ingredients", preventing a customer from conducting a comprehensive SCRM risk assessment, at the component level
- Current code signing practices are untrustworthy [3][4][5]; there is no standard, formal mechanism for a software customer to verify the trust relationship between a legitimate Software Supplier and the signing party on a digital signature applied to a software package, or SBOM, when available. This issue was confirmed by the CAB Forum code-signing work group on 6/4/2021 [2]
- As pointed out by NIST[1]; A solution is needed to "Identify and authenticate (I&A) trusted users: Implement an I&A scheme that adequately identifies the users who have the ability to sign code or submit code to be signed."; SBOM MUST provide the standard, formal method to identify a software supplier, at both a package and component level, which a software consumer can verify during an SCRM risk assessment, to achieve this goal.

The comments below are provided specifically to address SBOM Data fields, operational considerations, and support for automation, germane to a proactive (before software installation) SCRM cybersecurity risk assessment use case. Other use cases for SBOM may identify other requirements for NTIA consideration that are beyond the scope of this comment filing.

Pertaining to the Cybersecurity SCRM risk assessment Use Case

The comments below are based on the following assumptions:

- Software product(s) means: A commercially available, uniquely identifiable, named and versioned software product offering from a software vendor or developer that may be acquired, installed and operated by a software customer

- Software object means: Any artifact that may be used to perform a computing function within a computing device, as part of a software product
- Software package object means: A software object which a software vendor provides to a customer that is used to install a software product or patch release and its encapsulated objects
- Encapsulated object means: An artifact that is contained in a software package object
- Software application means: The full functionality provided by a software product through the combined contributions of encapsulated objects and software components required to make a software application function as expected. A single software product may contain multiple software applications
- Software component means: a software object that provides a defined set of functionality, which may operate independently or as part of another software object or software application
- Software products are, typically, delivered to customers as a "software package object", containing one or more encapsulated objects, some of which may contain software components that collectively provide software application functionality.
- A software package object serves as an "executable installation container", which software consumers execute to install all the software objects and artifacts required by a software product
- The software package object must be distinguishable from other encapsulated objects and software components within an SBOM to identify the "root of the software components hierarchy"
- After a successful software product installation, a software package object is no longer needed by a software customer, unless re-installation of a software product is required.
- Software package objects may contain several type of artifacts, referred to as an encapsulated object, including, but not limited to, application binary executables, license text, scripts, configuration and reference data, and other materials need to provide the complete set of application functionality expected by a software customer, from a software product
- Each "encapsulated object" contained in a "software package object" will have an appropriate entry in the "software package object SBOM" "ingredients list" that describes the baseline elements of each encapsulated object.
- An encapsulated object that is a software object, may itself contain one of more software components that were used in its construction
- An encapsulated object that is a software object may depend on another encapsulated software component from another software product, e.g. SSLEay32.dll form the OpenSSL product
- An SBOM MAY not include runtime platform dependencies or other external dependencies that exist outside of a software package object's "inventory list" of encapsulated objects.

Data Fields

SBOM Level Data Fields

- SBOM format identifier
- SBOM format version identifier
- Globally unique SBOM document identifier
- SBOM Document Name
- Author's Name
- Authors Identifier, using a formally recognized, standard URI
- Create Date and Timestamp in UTC

Software Package Object Data Fields

- Software Product Name (as assigned by the software supplier)
- Software Product Version (as assigned by the software supplier) – preferably semantic versioning nomenclature

- Software Product Source Supplier Name (Licensor of product) (name of the party that created the software package)
- Software Product Source Supplier ID, using a formally recognized, standard URI
- Software Package Object Name
- Globally unique Software Package Object Identifier
- Filename of the software package object installer file
- SHA256 File Hash of the installer file
- Software Package Build Date and Time in UTC
- Software Product Commercial Status (beta test, active production, retired, etc.)
- Software Product Support Status (commercial paid supported, open source community, unsupported, etc.)
- Authorized software package download location: URL where the software package file is available for download
- Embedded digital signature type: PKCS7 object, S/MIME, NONE, etc.
- Digital Signature File location: URL where the external digital signature file for this software package is located, blank if signature is embedded

Encapsulated Object Data Fields

- Object Type: text, binary, application, etc.
- Filename of the encapsulated object
- SHA256 File Hash of the encapsulated object
- Source download location: URL where filename of encapsulated object is available for download
- NON-SOFTWARE OBJECTS ONLY:
  - Source Supplier Name, name of the party that created the encapsulated object
  - Source Supplier ID, using a formally recognized, standard URI
- ENCAPSULATED SOFTWARE OBJECTS ONLY:
  - Software Source Supplier Name (Licensor of product) (name of the party that created the software package)
  - Software Source Supplier ID, using a formally recognized, standard URI
  - Software Product Name (as assigned by the software supplier)
  - Software Product Version (as assigned by the software supplier) – preferably semantic versioning nomenclature
  - Software Package Build Date and Time in UTC
  - Software Product Commercial Status (beta test, active production, retired, etc.)
  - Software Product Support Status (commercial paid supported, open source community, unsupported, etc.)
  - Embedded digital signature type: PKCS7 object, S/MIME, NONE, etc.
  - Digital Signature File location: URL where the external digital signature file for this software package is located, blank if signature is embedded

Operational Considerations

- SBOM artifacts MUST be digitally signed by a party authorized by the software source supplier of a software package object

- Software customers MUST have a defined, standard method to verify the trust relationship between an SBOM signer and the software supplier of a SBOM, using automation, as part of a digital signature verification process
- At a minimum, SBOM files and their digital signature files MUST be accessible via a publicly accessible website, using access control and encryption to protect the confidentiality of the SBOM data
- Alignment of SBOM data models and content with vulnerability repository service providers, enabling a software customer to search for vulnerabilities based on SBOM contents provided by a source supplier

Automation Support

- At a minimum SBOM's must be made available to software customers in the most common NTIA supported SBOM formats and canonical representations, SPDX Tag/Value and CycloneDX XML
- Identifiers used in an SBOM SHOULD utilize industry standard IANA registered URI schemes as identifiers, i.e. dns:softwareassuranceguardian.com for s software supplier identifier
- At a minimum SBOM MUST be encoded in the UTF8 format
- A method is needed to ensure that an SBOM and the software package which it describes exist in an immutable bonded trust relationship that cannot be altered without detection by a software customer

REA thanks NTIA for their consideration of these comments and looks forward to continuing to work with NTIA and other entities to help secure the software supply chain for all, with SBOM as the cornerstone.

.

REFERENCES:

[1] https://doi.org/10.6028/NIST.CSWP.01262018
[2] https://mailarchive.ietf.org/arch/msg/suit/ksrCCvtot-uSs_iTag48ASvE6sA/
[3] https://energycentral.com/c/ec/we-cannot-secure-software-supply-chain-without-sbom
[4] https://energycentral.com/c/ec/who-ya-gonna-trust
[5] https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf

Respectfully submitted by,

Dick Brooks
Co-Founder and Lead Software Engineer (CTO)
Reliable Energy Analytics LLC