Pacific Northwest
NATIONAL LABORATORY

PNNL-39652

Evaluation of Hardware and Software Bill of Materials (HBOMs/SBOMs) Extraction Methods

September 2026

William J. Hutton
Jessica L. Smith
Corinne G. Roth
Fleurdeliza A. de Peralta

U.S. DEPARTMENT of ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-39652

Evaluation of Hardware and Software Bill of Materials (HBOMs/SBOMs) Extraction Methods

September 2026

William J. Hutton
Jessica L. Smith
Corinne G. Roth
Fleurdeliza A. de Peralta

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Evaluation of Hardware and Software Bill of Materials (HBOMs/SBOMs) Extraction Methods

Dr. William Hutton, Dr. Jess Smith, Corinne Roth, Fleur De Peralta

Pacific Northwest National Laboratory
Richland, WA
will.hutton@pnnl.gov, jess.smith@pnnl.gov, corinne.roth@pnnl.gov, fleurdeliza.deperalta@pnnl.gov

***Abstract*—Hardware and software bills of materials (HBOMs and SBOMs) provide important visibility into the components, dependencies, and supply chain relationships within programmable digital devices. This visibility is critical for advanced nuclear reactor applications, where use of common or shared hardware components, software libraries, suppliers, or manufacturing processes may create common cause failure (CCF) vulnerabilities despite apparent diversity. This paper evaluates current approaches for obtaining and analyzing HBOMs and SBOMs in support of CCF, diversity and defense-in-depth (D^3) assessments, and begins to explore potential methods for artificial intelligence/machine learning-based analysis.**

The availability of BOM information from advanced reactor manufacturers and vendors, representative hardware and software categories found in advanced reactor systems continues to limit research [13]. This paper compares commonly used BOM formats, including CycloneDX, SPDX, and SWID. It also surveys publicly available tools for generating BOMs from source code, compiled binaries, and hardware-related information, noting limitations in language coverage, system age, and format interoperability.

Finally, this paper evaluates methods for correlating BOM data with vulnerability and exploitability information, including VEX, CVE, and CWE resources. The findings indicate that publicly available nuclear-vendor BOMs are limited, making third-party extraction and research into novel analysis techniques necessary.

***Index Terms*—Advanced reactors, cybersecurity, programmable digital devices, hardware bill of materials, software bill of materials**

I. Introduction

The Advanced Reactor Safeguards & Security (ARSS) program is a multi-lab effort funded by the U.S. Dept. of Energy. ARSS was established to provide research support to address near term challenges that advanced nuclear reactor vendors will face in meeting domestic Material Control and Accounting (MC&A), Physical Protection System (PPS), and cybersecurity requirements for U.S. construction [27].

The "AI/ML Models for CCF & D^3 Analysis" research project under ARSS is exploring how artificial intelligence (AI) and machine learning (ML) algorithms may be used for analysis of common cause failures (CCF) and diversity and defense-in-depth (D^3) issues based on hardware and software bills of material (HBOM and SBOM).

INFORMATION RELEASE NUMBER

II. Background

Modern programmable digital devices contain multiple hardware and software components and possess complicated supply chains. Supply chains for devices may be unique to a specific device, identical for multiple devices, or similar for multiple devices depending on the manufacturing processes and supplier sources. There can also be similarities across multiple product lines from a manufacturer, due to that manufacturer leveraging components across multiple lines for cost effectiveness.

To add more complexity to the issue, if we look at an entire system's BOM, built up of multiple devices each with their own SBOMs and HBOMs, we may find BOMs from different manufacturers that contain the same components. Common libraries like "stdli" in software underpin most modern software, and common integrated circuits (IC) like a basic memory unit are used across many manufacturers drawing from the same limited set of IC manufacturers.

Because of these overlaps in supply chains, vendor diversity in programmable digital devices is no guarantee of diversity and may hide CCF issues. Comparing programmable digital device HBOMs and SBOMs is a crucial step in assuring true diversity and defense-in-depth and performing CCF analysis. However, HBOMs and SBOMs may be viewed as proprietary or a competitive advantage by vendors and system designers, leading to repeat work to generate HBOMs and SBOMs for widely-used programmable digital devices in advanced reactors.

Without HBOMs and SBOMs, programmable digital devices may be opaque to subject matter experts. However, high-fidelity HBOMs and SBOMs may contain thousands of entries, making human analysis impractical. Therefore, it is critical we develop AI-assisted methods of evaluating BOMs for potential CCF or other related concerns.

A. Common Cause Failure (CCF)

In the nuclear industry "common cause failure" (CCF) refers to the simultaneous failure of two or more redundant components, systems, or safety functions due to a single shared cause or coupling mechanism. Failures occur when diverse elements, which are intended to provide independent layers of safety or reliability, are all affected by a shared root cause–such as a design flaw, environmental stressors (e.g., temperature, humidity, etc.), manufacturing defect, or human error. CCFs create positive dependency among components, which can critically undermine redundancy and defeat defense-in-depth strategies.

From the perspective of cybersecurity, unquantified risk of CCF can arise when different vendors use the same software libraries or chipsets resulting in a shared vulnerability.

Recognizing and mitigating CCFs is essential in nuclear probabilistic risk assessments and design reviews. Nuclear industry standards that specifically address CCF include NUREG/CR-5485 and IEC 62340.

B. Diversity and Defense-in-Depth (D^3)

In the nuclear industry, diversity and defense-in-depth are complementary design principles used to ensure that no single failure, error, or unforeseen event can compromise reactor safety. These concepts are deeply embedded in the safety philosophy [20] of the U.S. Nuclear Regulatory Commission (NRC) and have been applied to reactor design, operation, and emergency preparedness for decades [32]. Defense-in-depth seeks to provide multiple, independent layers of protection against accidents and radiological releases, while diversity reduces the likelihood that a common cause could disable multiple safety systems simultaneously. Together, these principles increase the resilience of nuclear power plants.

Defense-in-depth is based on the premise that safety should not rely on any single protective barrier. Instead, nuclear reactors employ successive layers of protection, including inherent design features, engineered safety systems, physical barriers, operating procedures, and emergency response capabilities [32]. For example, radioactive material is contained by multiple physical barriers, such as the fuel matrix, fuel cladding, reactor coolant pressure boundary, and containment structure. Likewise, critical safety functions such as reactor shutdown, decay-heat removal, and electrical power supply are supported by redundant and physically separated systems. The NRC describes defense-in-depth as a fundamental safety strategy that uses multiple safeguards to prevent accidents and to mitigate their consequences should prevention fail [1].

The concept of diversity addresses a different but related challenge: the potential for common-cause failures. A common-cause failure occurs when multiple systems fail simultaneously because they share a common vulnerability, such as a design defect, software error, environmental condition, or human error. Diversity mitigates this risk by implementing safety functions using different technologies, physical principles, equipment types, or actuation methods. For example, a reactor protection system based on digital instrumentation and control equipment may be supplemented by an independent and technologically distinct backup system capable of performing the same safety function. By ensuring that safety systems are not susceptible to the same failure mechanism [26], diversity reduces the probability that a single defect could defeat multiple layers of protection [24].

Modern reactor licensing reviews frequently evaluate diversity and defense-in-depth together, particularly for advanced digital instrumentation and control systems. NRC guidance requires designers to assess whether credible common-cause failures could impair reactor protection functions and whether sufficient diverse means exist to maintain safe operation or safely shut down the reactor. In this context, defense-in-depth provides multiple layers of protection, while diversity strengthens those layers by ensuring they do not share the same vulnerabilities [26]. The combined application of these principles is intended to provide high confidence that essential safety functions can be maintained even under unlikely or unforeseen conditions.

C. Hardware and Software Bills of Material (HBOM, SBOM)

A Hardware Bill of Materials (HBOM) and a Software Bill of Materials (SBOM) are structured inventories that document the components comprising a hardware or software product, respectively. Similar to a manufacturing parts list, these digital artifacts provide visibility into the constituent components of a system and the relationships between them.

An HBOM typically identifies physical components such as processors, memory devices, integrated circuits, firmware-bearing modules, suppliers, and part numbers. In the context of cybersecurity and supply chain risk management, HBOMs improve transparency regarding the provenance and composition of hardware products, enabling organizations to assess risks associated with counterfeit, vulnerable, or untrusted components. Recent U.S. government guidance has promoted HBOMs as a mechanism for strengthening supply chain assurance and supporting procurement decisions for critical technologies.

An SBOM is the software equivalent of an HBOM. The U.S. National Telecommunications and Information Administration (NTIA) and the National Institute of Standards and Technology (NIST) define an SBOM as a formal record containing details about the software components, dependencies, and supply chain relationships used to build a software product. An SBOM typically includes information such as component names, versions, suppliers, dependency relationships, authorship, and timestamps. By providing a machine-readable inventory of software ingredients, SBOMs enable organizations to identify affected systems when new vulnerabilities are disclosed, support license-compliance activities, and improve software supply chain transparency. These capabilities have become increasingly important following major software supply chain incidents and the U.S. government's broader cybersecurity initiatives under Executive Order 14028 [21].

Together, HBOMs and SBOMs provide complementary visibility into the hardware and software layers of modern information systems. When maintained throughout a product's

life cycle, they support vulnerability management, incident response, asset management, and supply chain risk assessment by enabling organizations to understand exactly what components are present in a system and where those components originated. As a result, BOM-based transparency is increasingly recognized as a foundational practice for cybersecurity, critical infrastructure protection, and secure technology acquisition [19].

III. Advanced Reactor Vendors Offering HBOMs & SBOMs

The obvious place to begin the search for HBOM and SBOM data would be advanced reactor vendors that make HBOMs and SBOMs publicly available. Unfortunately, it appears that no nuclear system vendors make any HBOM or SBOM data publicly available. This is an unfortunate lack of transparency, but is most likely due to competitive advantage in an emerging field, and not security by obscurity. Some advanced reactor vendors (e.g., GE) may provide BOMs to customers under signed non-disclosure agreements (NDAs).

A report by Idaho National Laboratory (INL) recommends that SBOMs be developed prior to the deployment of nuclear power plants [10]. INL's report also states that the generation of third-party SBOMs is often necessary because vendors do not typically provide SBOMs to customers (see Section VI).

An article in *Nuclear Technology* presents the challenges with generating SBOMs for typical nuclear systems [31]. Both reports underscores the lack of vendor provided BOMs.

IV. NuScale Power Module Control System (MCS)

The NuScale Power Module is a Small Modular Reactor (SMR) based on a pressurized water-cooled reactor technology system designed for electric power generation and various heating applications. It leverages the NuScale Power Module Control System for reactor monitoring and operations. The MCS is not a safety instrumentation system (SIS), but works with the SIS in system operations. The MCS is designed to provide centralized control for multiple reactors in different physical locations.

The following hardware and software categories were identified from the final safety report filed by NuScale Power [23]. As no BOMs are currently publicly available, we believe this basic set of components may provide the basic structure and modules from which to build a system level BOM. Many of these devices will contain both HBOMs and SBOMs, and coordination of the BOMs at the reactor/system level may prove challenging; a common format throughout the device BOMs and regular protocols to ensure up to date BOM data will be critical to ensuring security and integrity of the system.

It should be noted that the hardware and software categories included below may be implemented through devices which hold both hardware and software and therefore will have both (and possibly multiple) SBOMs and HBOMs. For example, engineering workstations will have an HBOM with the motherboard, processor, memory, etc, and multiple SBOMs for the operating system and each application installed on the operating system.

A. MCS Hardware Categories

- Distributed Control System (DCS) controllers
- Human-Machine Interface (HMI) workstations
- Engineering workstations
- Network communication equipment
- Input/Output (I/O) processes
- Remote interface units
- Plant data Historian servers
- Time synchronization equipment

B. MCS Software Categories

- Automatic reactor power control
- Startup/shutdown sequencing
- Control loop algorithms
- Alarm processing
- Historian (i.e. data logging) functions
- HMI display software
- Diagnostic and self-test software
- Configuration management software

V. HBOM and SBOM Formats

Some vendors make HBOMs and SBOMs available directly to customers. Unfortunately, these vendors are not publicly identified. To create useful HBOMs and SBOMs this project would need to work directly with vendors to identify the specific hardware and software produces and their versions. In absence of this information, this project could work with vendors that market similar hardware and software components that are not associated with a specific advanced reactor design for the purpose of researching the effectiveness of HBOMs and SBOMs as a crucial input into the AI/ML analysis of CCF and D^3.

There are three primary SBOM formats, CycloneDX, SWID, and SPDX. CycloneDX and SPDX are developing HBOM modules to add to the base SBOM format, but these are still in development at this time (July 2026). Foundational documents describing SBOM structure and formats come from both NTIA [21], NIST [19], and DHS CISA [5], while the foundational documents for HBOMs are from DHS CISA [7].

A. Minimum BOM Elements

Regardless of the specific format, a good SBOM as defined by all of the foundational documents listed above must have the following fields. A HBOM has a similar set, minus the version information. With this basic minimum set of elements, it becomes possible to compare BOMs generated by both different tools as well as in different formats.

- **Author**: The author of a BOM is the individual or entity that was responsible for the BOM generation. If the creator of the software publishes the BOM, this may also be documented in the "Software Producer" field.
- **Timestamp**: Software often changes, due to updates or configuration changes. It is critical to identify when the BOM was created. This often can be linked to the version number as well (see below).

- **Hash** (i.e. Unique Identifier: Each BOM is required to have a universally unique identifier (UUID) associated with the BOM itself. This can be a unique ID assigned by the author or it can be a hash of the software.
- **Component Name**: While the UUID is a machine-readable identifier, the component name is a human-readable identifier for the software or hardware component under evaluation.
- **Version Number**: An update to the software can significantly alter the BOM, relevant security concerns, the digital rights, or any other items of concern. A version number to assist with uniquely identifying the software component in a human readable manner is critical to ensure correct application of the BOM.
- **Relationships**: A BOM has two subareas of elements. The elements listed above and many of the optional elements describe the content of the software, which is the first of these subareas, but it is also critical to describe the other subarea, the relationships of the sub-components to both their parent component and to the other sub-components. How sub-components are connected can alter both the functionality and vulnerability of the overall system or device. In hardware, this is easily seen in that we need to know which integrated circuit is on which board or in which device in a system the USB port is enabled.

B. BOM Formats

The evaluation of the formats listed below should not be considered a recommendation for any of the formats. Each format was designed for a different use case and may be appropriate in different scenarios. Most of these formats are usually documented in a .json or .xml structure, but specific software may use a proprietary structure (i.e. a unique format outside of these completely). It is strongly suggested that any system owner intending to build their own BOM leverage a tool which is at least portable to one of these formats.

- **CycloneDX**: The CycloneDX format was designed for security and vulnerability management and asset tracking by OWASP [22].
- **SPDX**: The Software Packet iDentification eXchange format provides a mix of license management and asset/vulnerability management functionality. It was built by the Linux Foundation and supported by Microsoft. It is also an ISO standard (ISO/IEC 5962:2021) [15] [22].
- **SWID**: The SoftWare IDentification format is most commonly used for license management and tracking. It was developed by the National Institute of Standards & Technology (NIST) [22].

While CycloneDX, SWID, and SPDX are the most common formats, companies like Black Duck and government programs like DOE CESER's CyTRICS have alternative formats that are designed to enable specific features. For example, the CyTRICS format is designed to enable cybersecurity and integrity evaluations of control system equipment by researchers.

VI. HBOM and SBOM Extraction Methods

While some vendors do make SBOMs and HBOMs available, it may be necessary to develop BOMs for older systems or systems with vendors unable/uninterested in creating BOMs. For these systems, the following list of tools available to users may be helpful. Some tools require source code (i.e. pre-compiled instructions written by the vendor), others can reverse engineer binary files (also know as built or a build, made of compiled code, such as an application that you download). As highlighted by [29], a source BOM will look differently from a build BOM; the source will include named dependencies and variable names, and may include comments; a build BOM may include artifacts from compilation and may be more accurate to what is actually installed on a system.

Most tools will do a subset of programming languages, but no known tool works with all programming languages. Because of this, it may be necessary for a system owner/operator to leverage multiple tools. This is especially true in an industrial control system with a variety of vendors and ages of equipment.

Inclusion in the list below does not constitute recommendation of any tool. The list below is also not inclusive of every tool; there are many active efforts in this space and documenting every tool is not possible. For example, CycloneDX lists 296 different tools for different SBOM-related purposes. [9] There are also many custom tools, made by researchers or companies, which are not publicly available. The authors have included what appear to be the most common tools publicly available as of July 2026.

- **cdxgen**: Open source, CycloneDX format (see above), can create HBOMs and AI BOMs [25].
- **lynkctl**: Supports CycloneDX and SPDZ, intended for build pipelines, requires source code, designed for embedded C/C++ systems [14].
- **syft**: Open source, can parse binaries or source code, understands the Go language, built by Anchore [3].
- **Aikido**: Supports CycloneDX and SPDX, intended for build pipelines, requires source code [2].
- **Black Duck**: C-based, makes use of security containers, can export to CycloneDX or SPDX formats, uses proprietary BD format for management tools (suite of linked software tools) [4].
- **Finite State**: Can parse binaries or source code, supports CycloneDX and SPDX formats [11].
- **FOSSA**: Can parse binaries or source code, support for CycloneDX and SPDX formats, made by FOSSA [12].
- **Microsoft**: Available on github, SPDX format, support for multiple languages and Linux packages, continuous improvements [16].
- **Netrise**: Supports CycloneDX and SPDX, binary only, can parse many languages and Linux [17].
- **Snyk**: Supports CycloneDX and SPDX formats, designed to support source code build pipelines [28].
- **Trivy**: CycloneDX and SPDX formats, has packages for Linux variants, can parse many languages [30].

VII. BOM-Vulnerability Correlation

One of the most immediate ways in which a BOM can be useful is through asset and vulnerability management. The SBOM gives the end user the libraries or other dependencies in the software, allowing them to compare that list to the list of known vulnerabilities or weakness, without having to rely on information being monitored and then passed through the vendor or manufacturer. To accomplish this, various SBOM management tools compare the BOM to VEX, CVE, or more rarely CWE. It should be noted that, while a management tool will likely identify many vulnerabilities for a given BOM, this does not mean the software is vulnerable and should be disused. Vulnerability information needs to be evaluated by a human who understands the environment, use case, current version/status, and many other factors involved with the specific software deployment.

When considering use of BOMs in a nuclear facility to ensure security or integrity, it is possible to leverage these resources largely because of the overlap in control systems between nuclear and non-nuclear applications. Programmable logic controllers (PLCs), remote terminal units (RTUs), or other operational technology equipment are often made to be multi-purpose, connected to a wide range of meters, sensors, servos, and gauges. A device may be provisioned for a very specific task in a facility, but the underlying computing architecture is the same as that broader use case. Therefore, a vulnerability in a PLC may affect both a chemical manufacturing pressure vessel and a natural gas pipeline pumping station.

Similarly, the underlying components within a device, which should be listed in the SBOM and HBOM, are often very similar between control system devices and "normal" IT style devices. Therefore, the vulnerabilities discovered and reported in standard electronics can often be applied to a control system device. The impact of the vulnerability is often wildly different due to the implementations of the devices, but knowledge of the possibility allows the operator of the nuclear system to make an educated decision about their attack surface and risk profile.

A. VEX

The Vulnerability Exploitability eXchange was built by the makers of CycloneDX to interface with their BOM structure. It is designed as a primarily machine readable format [6].

B. CVE

The Common Vulnerability Exchange, maintained by MITRE, is a traditional vulnerability source. It provides specific vulnerabilities for given software as well as information such as the severity of impact. It is not meant to be machine readable [8].

C. CWE

The Common Weakness Enumeration, also maintained by MITRE, breaks down classes of weaknesses in software more than specific weaknesses in a specific piece of software. These can be useful for categorizing vulnerabilities within a BOM tool [18].

VIII. Conclusion

Despite specific HBOMs and SBOMs from programmable digital devices used in advanced reactors, standard tool sets and common file formats allow this research to continue using other HBOMs and SBOMs. Next steps will be to evaluate HBOM and SBOM data for its applicability to AI/ML algorithms, then finally, the effectiveness of AI/ML algorithms to perform CCF and D^3 analysis, which are cybersecurity issues common to any critical infrastructure, key resource, or fault-tolerant system. Continuing this research will position this approach to cybersecurity analysis for the nuclear industry when they are ready to apply it to their own HBOMs and SBOMs.

References

[1] Defense in Depth. [Online]. Available: https://www.nrc.gov/reactors/operating/ops-experience/fire-protection/defense-in-depth

[2] Aikido Security. Syft. [Online]. Available: https://github.com/anchore/syft

[3] Anchore. Syft by Anchore. [Online]. Available: https://anchore.com/syft/

[4] Black Duck. Black Duck SCA: SBOM Generation. [Online]. Available: https://community.blackduck.com/s/article/Black-Duck-SBOM-Generation-GUI

[5] CISA. SBOM FAQ. [Online]. Available: https://www.cisa.gov/resources-tools/resources/sbom-faq

[6] ——, "Vulnerability Exploitability eXchange (VEX) – Use Cases," Cybersecurity & Infrastructure Security Agency, Tech. Rep., April 2022.

[7] ——, "A Hardware Bill of Materials (HBOM) Framework for Supply Chain Risk Management," Cybersecurity & Infrastructure Security Agency, Tech. Rep., September 2023.

[8] CVE. CVE Program Mission. [Online]. Available: https://www.cve.org/

[9] CycloneDX. CycloneDX Tool Center. [Online]. Available: https://cyclonedx.org/tool-center/?ref=dependency-heaven

[10] S. Eggers, T. Simon, B. Morgan, E. Bauer, and D. C. (PNNL), "Towards Software Bill of Materials in the Nuclear Industry," Idaho National Laboratory, Tech. Rep., September 2022.

[11] Finite State. The Product Security OS for Connected Devices. [Online]. Available: https://finitestate.io/

[12] FOSSA. Best in Class SBOM Management. [Online]. Available: https://fossa.com/solutions/sbom-management/

[13] W. Hutton, C. Roth, and F. D. Peralta, "Programmable Digital Devices used in Advanced Reactors," Pacific Northwest National Laboratory, Tech. Rep., June 2026.

[14] Interlynk. Interlynk SBOM Generator. [Online]. Available: https://docs.interlynk.io/lynkctl/lynkctl

[15] ISO. Information Technology–SPDX Specification v2.2.1. [Online]. Available: ISO.org/standard/81870.html

[16] Microsoft. Microsoft SBOM-Tool. [Online]. Available: https://github.com/microsoft/sbom-tool

[17] NETRISE. Manage Supply Chain Risk with the Most Complete SBOM Solution. [Online]. Available: https://www.netrise.io/solutions/software-bill-materials-management

[18] NIST. NVD CWE Slice. [Online]. Available: https://nvd.nist.gov/vuln/categories

[19] ——. Software Security in Supply Chains: Software Bill of Materials (SBOM). [Online]. Available: https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-supply-chain-security-guidance-20

[20] M. D. (NRC), B. W. (NRC), J. L. (BNL), and V. M. (BNL), "Historical Review and Observatgions of Defense-in-Depth (NUREG/KM-0009)," Brookhaven National Laboratory, Tech. Rep., April 2016.

[21] NTIA. The Minimum Elements For a Software Bill of Materials (SBOM). [Online]. Available: https://www.ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom

[22] ——, “Survey of Existing SBOM Formats and Standards,” Natioanl Telecommunications and Information Administration, Tech. Rep., 2021.
[23] NuScale Power, “NuSale Final Safety Evaluation Report (ADAMS Accession: ML20023A318),” Nuclear Regulatory Commission, Tech. Rep., August 2020.
[24] R. W. (ORNL), R. (ORNL), M. C. (ORNL), D. H. (ORNL), K. K. (ORNL), A. L. (ORNL), G. M. (ORNL), M. M. (ORNL), J. M. (ORNL), I. O. W.P. Poore, A. Q. (ORNL), T. W. J. (ORNL), and M. W. (NRC), “Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems (NUREG/CR-7007, ORNL/TM-2009/302),” Oak Ridge National Laboratory, Tech. Rep., February 2010.
[25] OWASP. cdxgen. [Online]. Available: https://cdxgen.github.io/cdxgen/#/
[26] G. G. Preckshot, “Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems (NUREG/CR-6303, UCRL-ID-119239),” Lawrence Livermore National Laboratory, Tech. Rep., December 1994.
[27] SNL. Advanced Reactor Safeguards and Security. [Online]. Available: https://energy.sandia.gov/programs/nuclear-energy/safety-security-and-safeguards-for-advanced-nuclear-power/advanced-reactor-safeguards-and-security/
[28] Synk. Synk User Docs. [Online]. Available: https://docs.snyk.io/developer-tools/snyk-cli/snyk-cli/commands/aibom
[29] L. Tate, R. Jones, D. Dennis, T. Benko, and J. Askren, “Comparing Bills of Materials,” Pacific Northwest National Laboratory, Tech. Rep. [Online]. Available: https://www.pnnl.gov/sites/default/files/media/file/Comparing%20Bills%20of%20Materials%20Tate.pdf
[30] Trivy. Trivy v0.72 User Guide. [Online]. Available: https://trivy.dev/docs/latest/guide/supply-chain/sbom/
[31] H. Vairagade, A. Mercado, A. R. Padron, A. Buniya, S. Eggers, and F.Zhang, “Evaluating Methods of Software Bill of Materials Generation to Enhance Nuclear Power Plant Cybersecurity,” Nuclear Technology, Tech. Rep. 6, April 2025.
[32] T. Wellock. Defense in depth - a war for safety. Nuclear Regulatory Commission. [Online]. Available: https://www.nrc.gov/reading-rm/basic-ref/students/history-101/defense-in-depth

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov