Trail of Bits
497 Carroll St, Suite 71
Brooklyn, NY 11215

Trail of Bits's Response to PEO IEW&S Automated AIBOM RFI

About Trail of Bits: *Since 2012, Trail of Bits has helped secure some of the world's most targeted organizations and devices. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. We help our clientele—ranging from Meta to DARPA—lead their industries. Their dedicated security teams come to us for our foundational tools and deep expertise in reverse engineering, cryptography, virtualization, malware, and software exploits.*

About the authors: *Mr. Michael Brown is a Principal Security Engineer at Trail of Bits and specializes in the research and development of both conventional and AI-driven cybersecurity tools. His work, primarily for the US Department of Defense (DoD), ranges from performing in-depth security assessments of complex systems to creating tools for analyzing, hardening, and transforming software. He has successfully led several research programs funded by the Office of Naval Research (ONR) and the Defense Advanced Research Projects Agency (DARPA) that have developed novel cybersecurity capabilities including those driven by AI systems and models.*

Mr. Adelin Travers is a Senior Machine Learning Assurance Engineer at Trail of Bits working on major AI and ML risk engagements. His work exists at the intersection of ML, AI, and security, focusing on offensive, audit, and forensics methods as drivers of interpretable safety-critical AI. His past ML security and privacy research include Adversarial ML against audio systems and Machine Unlearning at the CleverHans lab, fingerprinting ML services and repurposing custom ML operators for code execution. He has contributed to ML security industry standards such as MITRE ATLAS and the NIST AI Risk Management Framework.

Dr. Heidy Khlaaf is the Machine Learning (ML) Assurance Engineering Director at Trail of Bits and specializes in the evaluation, specification, and verification of complex or autonomous (e.g., ML) software implementations in mission-critical systems, ranging from UAVs to large nuclear power plants. Her expertise ranges from leading numerous system safety audits (e.g., IEC 61508, DO-178C) that contribute to the assurance of safety-critical software within regulatory frameworks and safety cases, to bolstering the dependability and robustness of complex software systems through techniques that identify and mitigate system and software risks.

Trail of Bits commends PEO IEW&S for fostering an open discussion on the applicability of and methods for developing automated tools for Artificial Intelligence Bill Of Materials (AIBOM) through request for information (RFI). We offer recommendations informed by our expertise in cybersecurity, SBOM, AI/ML assurance, and safety auditing of mission-critical software.

Topic #1: Please provide general thoughts/recommendations along with any significant pros and/or cons to the proposed AIBOM concept

Trail of Bits views the AIBOM concept as a natural extension of existing Bill of Materials concepts currently used to document and audit the hardware and software components that make up our nation's critical infrastructure, warfighting platforms, communications systems, etc. Recent efforts to research and develop Software Bill of Materials (SBOM) tools, including open-source tools created by Trail of Bits such as pip-audit[1] and It-Depends[2], are now paying dividends for cybersecurity personnel. Such tools are being used to automate security scans for known software vulnerabilities and accelerate efforts to defend against newly discovered vulnerabilities. There is tremendous potential for the DoD to proactively ensure the safety, security, and performance of mission critical systems by adapting the SBOM concept to AI/ML (Machine Learning) models *before* they become pervasive.

However, we strongly caution against the adoption of AIBOM concepts that naively treat deployed AI/ML systems as specialized software components. While AI/ML systems are built, trained, and deployed using many of the same technologies as traditional software, they are fundamentally different approaches to solving problems (i.e., descriptive vs. prescriptive). Thus, a successful AIBOM concept (and in turn, AIBOM tools) must account for the unique aspects of AI/ML systems that go beyond those captured by SBOM tools.

Significant Pros to the AIBOM concept:

1. Extends familiar concepts (Hardware and Software BOM) for providing a deep understanding of a system's components and how they affect safety, security, and performance.
2. Can capture AI/ML system components that are missed by naively applying an SBOM tools:
    - Raw data sets
    - Data collection, curation, cleaning, transformation, and sampling processes
    - Sensor modules / hardware (for AI operating in physical domains)
    - Guardrail implementations
    - API/interfaces to traditional software
    - AI/ML model type, implementation, hyperparameters, probabilistic characteristics, algorithm, loss functions, etc.
    - Tools used for training (including standard model and optimization components implementation provided in ML frameworks)
    - Tools used for inference (these are often different from training due to configurations such as runtime and deployment platform)
    - Model compilers, transformers, formatters used for model portability and speed

Significant Cons to the AIBOM concept:

[1] https://github.com/pypa/pip-audit
[2] https://blog.trailofbits.com/2021/12/16/it-depends/

1. While the proposed AIBOM concept will enable improved security auditing over SBOM alone, there are several key aspects of model training and use that cannot be captured statically. As a result, the AIBOM concept cannot provide a complete security audit and must be complemented by other approaches (see response to Topic 2 below). Some notable examples include:
    - Controlling the order in which training data is ingested by the model is a potential data poisoning attack vector [3].
    - Customized code (e.g., lambda layers, operator redefinition) used in some models are a potential insertion point for malicious code.
    - Attackers controlling part of the model's inference-time behavior can modify models on the fly (i.e., dynamically loading new malicious weights)[4].
    - Model transformation procedures such as model quantization are vulnerable to model backdoors [5].
    - Considerations of specific data trade-offs, e.g., privacy versus processing time, and fitness functions.
2. The proposed AIBOM concept does not capture specialized hardware components (e.g., graphics processing units or GPUs) that are commonly used in deployed AI/ML systems. Such hardware may have unique vulnerabilities (e.g, data leakage). In addition to including a constituent SBOM, the AIBOM concept should be extended to include a Hardware Bill of Materials (HBOM).
3. The AIBOM doesn't cover third party prediction APIs (e.g. LLM applications based on OpenAI's model APIs and the more general Machine-Learning-as-a-Service paradigm).
4. The AIBOM by nature cannot attribute a prediction to a model (even if the model is subject to the AIBOM itself). Thus the provenance guarantees AIBOM offers are up to the model only and do not extend to downstream systems.
5. Novel structural vulnerabilities and supply chain intrusions arise due to the use of AI in the construction of downstream dependencies. New and undetectable attack vectors, such as poisoning web-scale training datasets and "sleeper agents" within large language models, may intentionally or inadvertently assist subversion of existing supply chain integrity.

Topic #2: Provide any alternatives to an AIBOM to address potential vulnerabilities in the supply chain of components that go into creating AI Models.

While we have noted that there are many significant cons to the AIBOM concept, it is worthwhile to note that many similar shortcomings also affect the use of SBOM as a software security auditing concept. As is the case with SBOM and other software security tools, Trail of Bits believes that AIBOMs are an important part of the AI/ML security ecosystem and should be complemented by other techniques to ensure strong model supply chain security.

[3] https://arxiv.org/abs/2104.09667
[4] https://arxiv.org/pdf/2004.11370
[5] https://arxiv.org/abs/2104.15129

As such, we urge the reader to consider the following techniques that address the AIBOM shortcomings we noted in Topic #1 as complements to the AIBOM concept rather than alternatives:

1. Data cleaning / normalization tools
2. Anomaly detection and integrity checks (checksum, hashes, etc.)
3. Data signing
4. Model and framework signing including for Ahead-Of-Time compiled models
5. Training and inference environment configuration verification
6. Use of safe model storage formats
7. Hardware specification for components like GPUs
8. Traditional infrastructure security for data pipelines
9. Tools for detecting anomalous data patterns

Topic #3: Provide recommendations on the proposed contents of AIBOM to address potential vulnerabilities in the AI Model supply chain.

First, we refer the reader to our list of AIBOM components enumerated in the second entry of our pros list in our response to Topic #1. We consider this list to be a set of concrete inclusions under the categories of "Model details" and "Data Lineage" in the RFI's proposed AIBOM concept.

Further, we recommend adding additional categories to the AIBOM concept to account for data and model transformation components. Examples of items tracked by these categories and the auditing functions they would enable follow:

1. The ability to trace initial raw data sets including data signing
2. Information about the labels including description of the labeling procedure (e.g., manual or automatic labeling)
3. Completeness, consistency, and correctness of data, specifically with consideration of unsupervised learning
4. Detailed information on data transformation procedures in the model pipeline (e.g., cleaning methods, normalization procedures, formulas, methods etc.)
5. A mechanism to track the evolution of the data as it goes through transformations to detect transformation and ordering based attacks
6. Model construction process, hyperparameters etc. as well as the relevant configurations of the frameworks at both train and inference time
7. Infrastructure security configuration for the data pipeline
8. Deployment infrastructure information including hardware, security and configuration adequation with the model training infrastructure

Topic #4:   Provide recommendations on technical means to implement and/or automate the components of the AIBOM in an AI/ML Ops pipeline.

Assuming that mature, automated AIBOM tools on par with existing SBOM tools are available, incorporating them into the AI/ML Ops pipeline would resemble how SBOM generation and auditing functions are currently integrated into the DevSecOps pipeline used in software engineering, albeit with several key differences.

The primary difference would involve interposing AIBOM tools on AI/ML specific stages that are controlled by the (potentially third-party) organization implementing the AI/ML Ops pipeline like data collection, data transformation, model creation, model configuration and model transformation. Due to the probabilistic nature of ML and as similarly pointed out for ML privacy[6], this is required to generate a complete AIBOM that can be meaningfully audited later for security issues.

To perform effective auditing, additional investment must be made into creating a suitable weakness database (similar to MITRE's CVE DB[7]) for AI/ML-specific components that are not considered software packages (e.g., model hyperparameters, transformation procedures. Previously proposed databases documenting instances of AI/ML vulnerabilities are unsatisfactory for the purpose of the AIBOM concept as they do not enforce a strong definition of AI/ML vulnerability. Rather, existing databases function more as a collection of AI/ML security and ethics related incidents. The proposed database should define a unique abstraction for AI/ML weaknesses and enforce a machine-readable format to allow it to be used as a data source for AIBOM security auditing.

Topic #5: Provide a list of tools, processes, and skills required to implement an AIBOM into an AI/MLOps pipeline.

Assuming that mature, automated AIBOM tools on par with existing SBOM tools are available, incorporating them into the AI/ML Ops pipeline would require tools, processes, and skills similar to those required to integrate SBOM generation and auditing functions into the DevSecOps pipeline. While we envision that AIBOM tools could be integrated into the AI/ML Ops pipeline by IT operations and cybersecurity personnel as is common with SBOM, it is important to note that such staff would require additional baseline knowledge on topics such as data science, data management, and specialized hardware used in AI/ML deployments. Further, we assume here that significant research advances have been made to automatically handle tracing of data lineage and provenance.

Topic #6: Provide an estimated cost for what it would take to produce an AIBOM.

Currently the cost to produce an AIBOM manually will vary greatly by system complexity. However, we argue that this cost is prohibitive for all but trivial AI/ML production systems due to the breadth of knowledge required to document a complete AIBOM. This is exacerbated by the fact that this knowledge is likely not entirely internal to the organization deploying the system due to third-parties involved (pre-trained model providers like OpenAI, dataset vendors, etc.).

Ideally, a mature set of automated AIBOM generation tools would allow cybersecurity professionals to create an AIBOM (including SBOM and HBOM as subcomponents) at negligible cost per system. However, to reach such an end state would require a significant investment to research and develop

[6] https://www.usenix.org/conference/usenixsecurity22/presentation/thudi
[7] https://cve.mitre.org/

quality AIBOM tools. Based on Trail of Bits’ expertise in building SBOM generation / auditing tools, conducting AI safety / security audits, and creating mature open-source software, we estimate that the cost of creating AIBOM tools to be on the order of $2,000,000 to $4,000,000. This estimate includes necessary research phases to define, design and prototype novel techniques needed as per topic #1-4 (such as the instrumentation of ML frameworks).

Note that our estimate above does not include the creation of a suitable AI/ML weakness database that we describe in our response to Topic #4. As a separate endeavor, we estimate creating, populating, and maintaining such a database to cost on the same order of magnitude as the AIBOM tools themselves.