Trail of Bits
85 Broad St, Fl 17
New York, NY 10004

Trail of Bits's Response to OSTP National Priorities for AI 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.*

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 the Office of Science and Technology Policy (OSTP) for fostering an open discussion on developing a national AI strategy through request for information (RFI) on policies to protect rights, safety and national security. We offer recommendations informed by our expertise in cybersecurity and safety auditing of mission-critical software.

Topic #1: What specific measures – such as standards, regulations, investments, and improved trust and safety practices – are needed to ensure that AI systems are designed, developed, and deployed in a manner that protects people's rights and safety? Which specific entities should develop and implement these measures?

We believe that constructing and assessing verifiable claims for AI-based systems, to which developers can be held accountable, is a crucial step in helping protect people's rights and safety. Claims are assertions put forward for general acceptance that must be substantiated by evidence or arguments. The scope of a claim should be relevant to a regulatory, safety, ethical, or technical application, and must be sufficiently precise to be falsifiable. That is, a claim may hold true only within the boundaries of that scope, which must be specified.

In safety-critical and defense domains, claim-oriented or goal-based approaches have been consistently used to structure arguments regarding the safety of engineered systems, including autonomous and AI-based systems[1]. These approaches are predominantly known as safety or assurance cases. A safety case is a documented body of evidence that provides a convincing and valid argument regarding a top-level claim (such as the safety of an autonomous vehicle as defined in UL 4600[2]), and presents a structured justification in support of that claim to decide the status of it. Safety cases are often required as part of a regulatory process. For example, the FDA requires infusion pump manufacturers to submit safety cases as part of the 510(k)s.

A certificate of safety or a license is then granted only when a regulator is satisfied by the argument presented in a safety case. The goal-based approach and fluidity of a safety case allows licensees to determine the assurance activities that must be carried out in accordance with a regulator's safety goals or principles. Licensees are then responsible for ensuring that their use of a technology complies with these principles by conducting or commissioning assessments of their systems. This process leads to a documented formal qualification of a system for its intended application, backed by evidence, that can be presented to a regulator.

When determining the safety justification of software-based systems within a safety case, it is typically split across two stages: production excellence (i.e., accountability-by-design), in which the quality of the design and development processes is assessed, and independent assessment, which requires a thorough, independent examination of the device and/or its software. Production excellence is typically assisted by evidence of the systematic application of national and international standards (i.e., prescriptive approaches). IEC 61508, UL 4600, IEC 61513, and DO-178C are typical of the standards recommended for this role. We refer to these standards for a detailed review of records and other documentation required for systems to support AI accountability, as we believe that AI systems should be categorized as an extension of software-based systems given the identical mechanisms of their development. Current AI-based systems do not possess any unique software

[1] Bloomfield, R., Khlaaf, H., Ryan Conmy, P., and Fletcher, G., "Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy". Computer, vol. 52, no. 9, pp. 82-89, Sept. 2019, doi: 10.1109/MC.2019.2914775.

[2] "UL 4600: Standard for Evaluation of Autonomous Products, Edition 3". Underwriters Laboratories, March 2023.

components that warrant a generalized licensing scheme that would not heavily impede the use of software as a whole. Indeed, any implementation of such a scheme would likely result in significant overreach due to the broad definition and software components of AI systems. A further literature review on accountability mechanisms, including regulation and assessments, required for software-based systems across numerous safety-critical domains can be found in [3].

We believe the above processes would be too rigorous for non-mission-critical AI applications, and that AI regulatory policies should generally mirror the practices of existing sectors in which they are deployed. This includes inputs to audits or assessments and mandating accountability measures, including compliance with existing regulatory standards and practices throughout a system's lifecycle to provide assurance of the final design. Overall, AI accountability policies and regulations should largely be sectoral, and further regulation should be defined for novel domain areas where AI may produce novel harms (e.g., bias and discrimination in facial recognition or human resourcing). We believe that by defining a more concrete operational envelope (e.g., through a sector-specific and AI-based Operational Design Domain[4]), developers and regulators can better assess potential risks and required safety mitigations for AI-based systems.

Topic #5: How can AI, including large language models, be used to generate and maintain more secure software and hardware, including software code incorporating best practices in design, coding and post deployment vulnerabilities?

The modern software development and maintenance life cycle (SDLC) is complex and multi-faceted. It consists of several stages, each with numerous subtasks dedicated to designing, implementing, integrating, testing and deploying functional software. Adherence to secure coding practices and ensuring that appropriate security controls are properly implemented requires effort and discipline throughout the SDLC, not just when the code is written.

Current applications of general large language models (LLMs) for software development such as Copilot and Codex have largely aimed to generate code from prompts provided by software developers. Unfortunately, this use case for LLMs has dangerous implications for software security. A recent study observed that Copilot generates code with vulnerabilities approximately 40% of the time[5] due to the prevalence of bugs and vulnerabilities in the code used to train the LLM, a manifestation of "garbage in, garbage out". Whether they are deployed as part of the SDLC or as a replacement for it, LLM generated code must be closely inspected for latent bugs and vulnerabilities, ultimately making general LLMs poorly suited for secure code generation. Some attempts to employ smaller LLMs specialized to specific code generation applications[6] have shown some early success, but such models are challenging to obtain training data for, build, and deploy, ultimately limiting their impact on general software development.

---

[3] Butler, E., Fletcher, G., George, S., Guerra, S., and Khlaaf H., "Cots Digital Devices In Safety Critical Industries – Use and Licensing". Energiforsk AB, November 2019, ISBN 978-91-7673-627-2.
[4] Khlaaf, H., "Toward Comprehensive Risk Assessments and Assurance of AI-Based Systems". Trail of Bits, 2023. https://www.trailofbits.com/documents/Toward_comprehensive_risk_assessments.pdf.
[5] https://arxiv.org/pdf/2108.09293.pdf
[6] https://leandojo.org/

Despite serious concerns with using LLMs to directly generate code, more promising applications of LLMs have been suggested for subtasks in the SDLC[7]. Tasks such as suggesting changes to existing code to make it more readable, summarization of code in natural language to help new developers become familiar with a codebase, predicting build errors, offering code completion suggestions, etc. are well suited to LLMs that excel at natural language processing / generation. Still, due to the propensity of LLMs to confidently provide inaccurate or dangerous[8] responses, deployments of LLMs for use in SDLC subtasks must still be audited and used with appropriate guardrails to ensure safe and secure code is produced. As suggested by Khlaaf et al.[9], a base level of specialized knowledge (i.e., expertise in cybersecurity) may be required to use code generation models in order to distinguish between correct or frequently incorrect solutions, as the output of these models do not guarantee any sound or complete results regarding synthesis, generation, summarization, or other uses. Furthermore, over-reliance and over-trust on the model to generate mission-critical output (e.g., documentation or comments) may lead developers to miss implementation and safety relevant details that would otherwise be observed by manual processes.

While LLMs have received intense attention over the last year, other emerging AI approaches for improving software security have demonstrated success. In particular, recent research developed under DARPA Artificial Intelligence Exploration (AIE) programs[10] [11] has demonstrated that AI-based vulnerability scanners that use graph-based models outperform traditional vulnerability scanners for certain classes of cyber vulnerabilities[12]. Such techniques exploit the inherently graph-like structure of software and availability of well-structured test sets developed by NIST[13] to identify vulnerable software conditions that typically rely on human experts to detect. Additionally, recent efforts funded by the Office of Naval Research[14] are exploring the use of AI-based systems to automatically remove unnecessary code (referred to as “bloat”) from programs that may contain latent vulnerabilities.

Topic #6: How can AI rapidly identify cyber vulnerabilities in existing critical infrastructure systems and accelerate addressing them?

In general, finding and remediating vulnerabilities in critical infrastructure system software is a largely manual process. Many open-source and commercial software analysis tools (e.g., source code scanners, automated testers) have been built to help system developers and security teams identify potential vulnerabilities, however in virtually all cases the alerts produced by these systems must be manually evaluated to eliminate false positives. Due to the inherent limitations of software analysis, the point of diminishing returns has already been met in the research and development of

---

[7] https://ai.googleblog.com/2023/05/large-sequence-models-for-software.html
[8] https://www.deepmind.com/publications/ethical-and-social-risks-of-harm-from-language-models
[9] https://arxiv.org/abs/2207.14157
[10] https://www.darpa.mil/program/artificial-intelligence-mitigations-of-emergent-execution
[11] https://www.darpa.mil/program/recovery-of-symbolic-mathematics-from-code
[12] https://www.usenix.org/conference/usenixsecurity23/presentation/mirsky
[13] https://samate.nist.gov/SARD/test-suites/112
[14] https://www.navysbir.com/n23_A/N23A-T009.htm

these tools. New advances are few and far between and require high levels of effort to achieve. As a result, there remain entire classes of software vulnerabilities that rely on human experts to detect, assess, and remediate .

Despite being relatively under-researched, AI-based systems to identify cyber vulnerabilities in programs have presented a compelling alternative to traditional program analysis techniques because they are not subject to the same limitations. Further, false positive rates for AI-based systems can be improved by part by curating better training datasets over time, as opposed to expending considerable R+D funds to make marginal advances in analysis sophistication as is the case with traditional techniques. Finally, AI-based systems are efficient and scalable enough to complement traditional approaches, providing security teams with a best-of-both worlds approach. By employing both traditional and AI-based vulnerability detection systems, security teams can accelerate the vulnerability discovery and remediation process.

Unfortunately, modeling software in a manner that AI systems can learn to reason about is fraught with pitfalls due to the wide variety of AI modeling techniques available to tool developers. As is the case with any AI system, it is critically important to select an appropriate model for the problem to be solved. Generative models, such as the popular ChatGPT large language model (LLM) have received a significant amount of attention over the last year, but are ultimately poorly suited for vulnerability discovery and remediation, particularly when dealing with previously unknown or novel vulnerabilities[15].

LLMs are tailored for natural languages (e.g., chatbots), which are inherently different from computer languages despite the relative readability of modern source code. Due to their incredibly large training data sets (i.e., large swaths of the internet including source code repositories), large scale LLMs may appear to be capable of identifying vulnerabilities because they have been trained on countless articles and examples describing vulnerabilities in source code. However, when controlling for the source of the test case (i.e., is the test case likely to be present in the model's training data) it is clear that LLMs have accuracy on par with or below random guessing when asked to find vulnerabilities in programs that are not publicly available and discussed on the internet. As a result, it is likely that the emergent capabilities of LLMs to scan for vulnerabilities in programs is due to memorization, a common problem in AI systems.

Despite the shortcomings of generative models, other AI systems have shown great promise for identifying cyber vulnerabilities. In particular, recent research developed under DARPA Artificial Intelligence Exploration (AIE) programs[16] [17] has demonstrated that AI-based vulnerability scanners that use graph-based models outperform traditional vulnerability scanners for certain classes of cyber vulnerabilities[18]. Such techniques exploit the inherently graph-like structure of software and

[15] https://blog.trailofbits.com/2023/03/22/codex-and-gpt4-cant-beat-humans-on-smart-contract-audits/
[16] https://www.darpa.mil/program/artificial-intelligence-mitigations-of-emergent-execution
[17] https://www.darpa.mil/program/recovery-of-symbolic-mathematics-from-code
[18] https://www.usenix.org/conference/usenixsecurity23/presentation/mirsky

availability of well-structured test sets developed by NIST[19] to identify vulnerable software conditions that typically rely on human experts to detect.

It is important to note that the use of AI-based vulnerability detection systems should complement the use of existing tools. AI systems may be limited in their application if sufficient training data is not available. This shortcoming is counterbalanced by traditional approaches, which in turn are counterbalanced by AI systems' ability to address loosely-defined security problems that traditional approaches struggle with. There remains tremendous potential to improve the performance and capability of AI methods through additional research and development. As such, we urge the US Government to more aggressively fund the development of these techniques in order to maintain technological superiority with respect to cybersecurity.

Topic #7: What are the national security risks associated with AI? What can be done to mitigate these risks?

AI systems, in particular generative large language models (LLMs) such as Codex and ChatGPT, have demonstrated the potential to lower the technical expertise required to carry out cyber attacks. Such emergent capabilities present a clear risk to national security. For example, attackers can use ChatGPT to craft sophisticated phishing attacks such as spear-phishing and whaling with significantly less background research and far less effort than before. Further, these AI-generated attacks are significantly harder to detect because they do not contain misspellings and broken English grammar common to manually crafted social engineering messages. Similarly, other generative AI systems for audio/visual media such as Stable Diffusion have demonstrated the capability to generate convincing, images, audio streams, and videos that can be used to carry out psychological operations, extortion, social engineering and disinformation campaigns[20].

Of deeper concern is the potential for AI systems to reduce the technical expertise required for adversaries to find and exploit vulnerabilities in software. While initial research conducted by Trail of Bits[21] and the cybersecurity community at large indicates that LLMs struggle to identify and novel vulnerabilities in software, there is potential for advanced or specialized models to be used by attackers to rapidly develop or customize exploits against known (i.e., publicly disclosed) vulnerabilities. If deployed, such models would provide low-sophistication attackers with the speed and scale of action they need to exploit vulnerable systems before they are patched, a particular concern for our nation's critical infrastructure systems.

The first step in mitigating the threats to national security posed by AI systems is to fully understand their capabilities. Today's evaluation methods are primarily driven by manual experimentation and no effective systematic evaluation methods to assess the emergent cyber capabilities of AI systems currently exist. We urge the US government, Academia, and Industry stakeholders to invest in developing and unifying techniques for quantifying AI system capabilities under a systematic

[19] https://samate.nist.gov/SARD/test-suites/112

[20] https://knightcolumbia.org/content/how-to-prepare-for-the-deluge-of-generative-ai-on-social-media

[21] https://blog.trailofbits.com/2023/03/22/codex-and-gpt4-cant-beat-humans-on-smart-contract-audits/

evaluation framework. Such a framework is necessary to fully assess the cybersecurity risks AI systems may have on national safety and security and take steps to mitigate them.

It may seem counterintuitive to carry out a capabilities evaluation before a risk assessment. However, traditional risk assessments require implicit assumptions and knowledge regarding a prospective system's capacities, limitations, and failure modes (which in turn inform possible harms a system may pose) [22]. In the case of LLMs, for example, and more generally, generative AI, these capabilities and failure modes are not yet fully understood. The acceptance or mitigation of identified hazards and harms within a risk assessment must be evaluated based on performance criteria that define the tolerable risk allowed. Emergent capabilities that may have national security and safety implications thus require further evaluation regarding the complexity of security-related specifications and properties that LLMs can excel against.

Regarding scoping of national security risks, it is pertinent to assess model capabilities through application-specific evaluation benchmarks to inform risk assessments. In [23], we thus propose the integration of Operational Design Domains (ODDs) as first introduced by the National High Traffic Safety Administration (NHTSA)[24] into a risk framework, where we define a novel ODD taxonomy relevant to the use of AI technologies, including generative models. The purpose of an ODD is to describe the specific operating conditions under which an AI-system is designed to properly behave, thus outlining the safety envelope against which system hazards and harms can be determined.

Further investments must also be made to develop countermeasures to novel risks posed by AI systems. Research and development programs such as DARPAs MediFor and SemaFor[25] [26] projects have demonstrated success in countering deepfake technology. Similar programs are necessary for researching and developing countermeasures against AI systems. For example, novel watermarking techniques for LLMs[27] may be useful for attributing exploits and malware to specific models, providing a deterrent to adversaries seeking to use LLMs to exploit software. Failure to fund timely, well-directed research and development programs in this area presents an opening for adversaries to rapidly close the gap between their cyber capabilities and those of the US.

---

[22] Khlaaf, H., Mishkin, P., Achiam, J., Krueger, G., & Brundage, M., "A hazard analysis framework for code synthesis of large language models". arXiv. http://arxiv.org/abs/2207.14157

[23] Khlaaf, H., "Toward Comprehensive Risk Assessments and Assurance of AI-Based Systems". Trail of Bits, 2023. https://www.trailofbits.com/documents/Toward_comprehensive_risk_assessments.pdf.

[24] "A Framework for Automated Driving System Testable Cases and Scenarios". National Highway Traffic Safety Administration. DOT HS 812 623. https://rosap.ntl.bts.gov/view/dot/38824.

[25] https://www.darpa.mil/program/media-forensics

[26] https://www.darpa.mil/program/semantic-forensics

[27] https://arxiv.org/abs/2301.10226