NATIONAL SCIENCE FOUNDATION
2415 EISENHOWER AVENUE
ALEXANDRIA, VIRGINIA 22314

NSF 23-149

Dear Colleague Letter: Inviting Proposals Related to Open-Source Software Security to the Secure and Trustworthy Cyberspace Program

September 12, 2023

Dear Colleagues:

Open-source software (OSS) is pervasive in commercial products, government systems, and military platforms. Although this philosophy of software development accelerates and catalyzes innovation via composability, it is not without security risks[1]. Adversaries can leverage the pillars of the OSS philosophy - the democratization of software development and software reuse - to insert and exploit vulnerabilities in OSS code, especially when the code is written in memory-unsafe programming languages, such as C and C++.

Some reports suggest as many as 70% of all software vulnerabilities arise from memory safety issues[2]. Despite these risks, memory-unsafe programming languages remain pervasive in the software development ecosystem in part due to the mature and well-integrated set of methodology and tools that support developer productivity.

Even programs written in memory safe languages can be compromised because code reusability and modularity introduce dependencies, complexity, and liabilities to the software development life cycle. Furthermore, the dynamics of diverse organizations and teams of developers with different roles contributing to different projects pose unique challenges in the creation and maintenance of a secure OSS ecosystem.

Thus, securing the OSS ecosystem requires approaches that encompass the following facets: (i) technical aspects of software development, (ii) incentive and organizational structures to secure the OSS ecosystem, and (iii) educational research approaches for the education and training of the next generation of software developers who are knowledgeable in secure and memory-safe software development.

The purpose of this Dear Colleague Letter (DCL) is to encourage the submission of novel and high impact proposals, within or across disciplines, applying the highest

standards of research methodology and use of evidence, to advance knowledge on securing the OSS ecosystem, targeting at least one of the following areas/topics:

- **Software engineering frameworks/tools/methodologies** for efficient, usable, and secure software development to ensure memory safety: new memory-safe programming languages, extensions to existing memory-safe languages, and software engineering methods that support the development of memory safe software systems including secure type systems, secure compilation and optimization, static and dynamic analysis, fuzz testing, and formal verification.
- **Handling unsafe legacy code:** automating the transition of software/projects written in memory-unsafe to memory-safe programming languages.
- **Dependency management:** tracking and automatic updating code dependencies, especially when components are written in more than one language or include components written in unsafe languages, generation of Software Bill of Materials (SBOM), metrics, best practices, and auditing.
- **Trust and safety:** detection and mitigation of vulnerable and/or malicious commits, bad actors, and suspicious/vulnerable team dynamics, including approaches to understand, prevent, and discourage unsafe behaviors.
- **Incentive and organizational structures for a secure OSS ecosystem:** models for incentivizing safety and disincentivizing practices that lead to insecure code, including developer compensation considering their different roles in projects, incentives for organizations to transition unsafe legacy code to memory-safe programming languages and for developers/teams to operate with security and memory safety as first-class citizens in their daily tasks, accountability, and metrics for success.
- **Education and workforce development:** development of effective educational research approaches for teaching secure software development in formal or informal settings, including learning about the need and use of memory-safe programming languages.

This DCL does not constitute a new competition nor a new program. Rather, interested proposers should prepare and submit proposals in accordance with the instructions in the Secure and Trustworthy Cyberspace (SaTC) program solicitation. We invite submissions to the CORE, EDU and TTP designations of the SaTC solicitation as appropriate to the work proposed. See the SaTC solicitation for more details on these types of proposals, as well as the topics that the SaTC program is interested in around OSS Security. Additionally, to call attention to responsiveness to this DCL, project summaries should include "OSS Security" in the keyword list. Proposals submitted in response to this DCL will count towards the proposal limits imposed in the SaTC solicitation.

Questions should be directed to: satc@nsf.gov

Sincerely,

Margaret Martonosi, Assistant Director, Directorate for Computer and Information Science and Engineering (CISE)
Sylvia M. Butterfield, Acting Assistant Director, Directorate for Social, Behavioral and Economic Sciences (SBE)
James L. Moore III, Assistant Director, Directorate for STEM Education (EDU)
Sean L. Jones, Assistant Director, Directorate for Mathematical and Physical Sciences (MPS)
Susan S. Margulies, Assistant Director, Directorate for Engineering (ENG)

REFERENCES

1 Recommendations from the Workshop on Open-source Software Security Initiative. Angelos D. Keromytis. U.S. Open-Source Software Security Initiative Workshop, August 24-25, 2022. https://bpb-us-w2.wpmucdn.com/sites.gatech.edu/dist/a/2878/files/2022/10/OSSI-Final-Report.pdf
2 Catalin Cimpanu. Microsoft: 70 percent of all security bugs are memory safety issues. February 11, 2019. ZDNet https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues/