Small Business Innovation Research (SBIR) and
Small Business Technology Transfer (STTR)
Opportunity Announcement
HR001120S0019-01
Programming Language Support for Assured Data Intent

Which program will fund this topic?
STTR

What type of proposals will be accepted?
Phase I Only

Technology Area(s): Information Systems

I. INTRODUCTION
The Defense Advanced Research Projects Agency (DARPA) Small Business Programs Office (SBPO) is issuing an SBIR/STTR Opportunity (SBO) inviting submissions of innovative research concepts in the technical domain(s) of Information Systems. In particular, DARPA is interested in understanding the feasibility of Programming Language Support for Assured Data Intent.

This SBO is issued under the Broad Agency Announcement (BAA) for SBIR/STTR, HR001120S0019. All proposals in response to the technical area(s) described herein will be submitted in accordance with the instructions provided under HR001120S0019, found here: https://beta.sam.gov/opp/b8abeb02f16a4450b2c2f859fc00c177/view.

a. Eligibility
The eligibility requirements for the SBIR/STTR programs are unique and do not correspond to those of other small business programs. Please refer to Section 3.1, Eligible Applicants, of HR001120S0019 for full eligibility requirements.

b. Anticipated Structure/Award Information
Please refer to Section 1, Funding Opportunity Description, in HR001120S0019 for detailed information regarding SBIR/STTR phase structure and flexibility. Phase II program description and metrics are provided for informational purposes only. Proposers awarded a Phase I contract will be eligible to submit a proposal for Phase II and will be contacted by the DARPA Small Business Programs Office at the appropriate time during their Phase I period of performance.

For this SBO, DARPA will accept Phase I proposals for cost of up to $225,000 for a 7-month period of performance.

Proposers should refer to Section 4, Application and Submission Information, of HR001120S0019 for detailed proposal preparation instructions. Proposals that do not comply with the requirements detailed in HR001120S0019 and the research objectives of this SBO are considered non-conforming and therefore are not evaluated nor considered for award.
Phase I proposals shall not exceed 20 pages. Phase I commercialization strategy shall not exceed 5 pages. This should be the last section of the Technical Volume and will not count against the 20-page limit. Please refer to Appendix A of HR001120S0019 for detailed instructions on Phase I proposal preparation.

c. Evaluation of Proposals
Section 5, Evaluation of Proposals, in HR001120S0019 provides detailed information on proposal evaluation and the selection process for this SBO.

d. Due Date/Time
Full proposal packages (Proposal Cover Sheet, Technical Volume, Price/Cost Volume inclusive of supporting documentation, and Company Commercialization Report) must be submitted via the Department of Defense (DoD) SBIR/STTR Proposal Submission website per the instructions outlined in HR001120S0019 no later than 2:00 pm ET, April 20, 2020.

II. TOPIC OVERVIEW
a. Objective
Develop non-burdensome approaches for programmers to express intended uses of data held in computer memory and systematically enforce these intended uses throughout program run time.

b. Description
DoD has a critical need for protecting sensitive data to make sure that only the intended, trusted programs or parts of programs can access the data. This protection must happen not only when data is at rest (i.e., stored in files) but also when it is loaded into memory while being processed by software. Recent discoveries of high-bandwidth side channels across all types of modern CPUs revealed the staggering insufficiency of current approaches to this security goal and necessitate a new approach.

Specifically, baseline confidentiality protections of data contexts rely on programming language and operating system abstractions that did not require programmers to label data and code units. The default level of isolation that these abstractions provided was, until recently, deemed efficient, and additional annotations of code and data unit relationships were deemed unnecessary. Additional enforceable annotations, such as SELinux policies, were at the level of individual files for data and of processes for code.

However, the label-free approach baseline is no longer sufficient, as compelling recent research asserts that modern CPU side-channels defeat all default abstraction-based confidentiality expectations, and furthermore, that no comprehensive general software mitigations are likely to be discovered. [1] Thus, to be meaningfully protected, sensitive data and parts of the program must be singled out and treated specially in the program's structure. Meanwhile, although hardware memory protection mechanisms, such as encrypted memory, [2] have technological promise, programmers cannot easily take advantage of them, as leveraging these mechanisms requires both mastering complex
systems interfaces and significant changes to the structure of programs. On the programming language side, current type systems do not map to the level of Application Binary Interface (ABI) objects and concepts (such as hardware-tagged, protected memory regions) well enough for effective runtime enforcement.

This effort will explore approaches to protecting data and program parts on current and future CPUs and operating systems via non-burdensome, application-specific, programmer-added annotations of the relevant sensitive data and code units, and of their intended relationships at levels more granular than files and processes. The annotations must be expressible via Application Binary Interface (ABI) so that compilers, linkers, and operating systems will act on these annotations to automatically generate and apply specific isolation measures such as physical rather than logical separation of computations, hardware memory encryption applied to specific program contexts, etc., automatically refactoring the program as necessary.

To be non-burdensome, the annotations must be concise and largely inferable automatically from a few key places in the code, similar to modern type systems. Importantly, since sensitive data can be compromised via code not intended to access it, and also sensitive code can be exploited by data that it was not intended to access, annotations must apply to both code and data units to be effective.

The effort will produce design patterns for annotation types and program designs to be adopted by strictly-typed, memory-safe programming languages as a part of their type system expressing data policy properties, to be enforced systematically at runtime.

c. Phase I
Develop prototype code annotations and data unit annotations. Specify their intended relationships across a broad selection of use cases and programming patterns where security is associated with isolation or specific access patterns between code and data. Create a prototype demonstration to show how a programming language might support fine-grained, intra-process, runtime-enforced data isolation policies. Use the prototype to demonstrate feasibility of non-burdensome annotation for common programming tasks and program designs, such as data parsing-and-processing pipelines.

i. Schedule/Milestones/Deliverables – Phase I payable milestones for this SBO should include:
   - Month 2: Report documenting code annotations and data unit annotations for the prototype system.
   - Month 4: Prototype demonstration of fine-grained, intra-process, runtime-enforced data isolation policies.
   - Month 7: Prototype feasibility demonstration, and Phase I Final Report containing the annotation system design.

d. Phase II
Develop automated program analyses for reasoning about isolation properties, protection properties, and policy models. Develop ABI-level runtime mechanisms and refactoring
tools to implement prototypes developed in Phase I. Minimize performance impact of these mechanisms and policies.

This solicitation requests the development and demonstration of a proof of concept prototype of practical approaches for programmers to express intended uses of data held in computer memory, and systematically enforce these intended uses throughout program runtime, by the end of Phase II.

i. **Schedule/Milestones/Deliverables** – Phase II payable milestones for this SBO should include:
   - Month 2: Technical and financial status update
   - Month 4: Technical and financial status update
   - Month 6: Report on final design of annotation system; technical and financial status update
   - Month 8: Technical and financial status update
   - Month 10: Technical and financial status update
   - Month 12: Report on status of demonstrating the selected annotation system; technical and financial status update; prototype software delivery
   - Month 16: Technical and financial status update
   - Month 18: Technical and financial status update
   - Month 20: Technical and financial status update
   - Month 22: Technical and financial status update
   - Month 24: Phase II Final Report on finished annotation system; demonstration results; final software delivery.

e. **Dual Use Applications (Phase III)**
   Scale the mechanisms and methods developed in Phase II to a modern, large-code base such as a web browser. Transition developed solutions to DoD and commercial customers, as each have similar challenges with respect to maintaining the integrity of their cyber computing environments.

f. **References**


g. **Keywords**
   Cybersecurity, programming languages, security policies
III. SUBMISSION OF QUESTIONS

DARPA intends to use electronic mail for all correspondence regarding this SBO. Questions related to the technical aspect of the research objectives and awards specifically related to this SBO should be emailed to HR001120S0019@darpa.mil. Please reference BAA HR001120S0019-01 in the subject line. All questions must be in English and must include the name, email address, and the telephone number of a point of contact.

DARPA will attempt to answer questions in a timely manner; however, questions submitted within seven (7) calendar days of the proposal due date listed herein may not be answered. DARPA will post a consolidated Frequently Asked Questions (FAQ) document. To access the posting please visit: http://www.darpa.mil/work-with-us/opportunities. Under the HR001120S0019-01 summary, there will be a link to the FAQ. The FAQ will be updated on an ongoing basis until one week prior to the proposal due date.

In addition to the FAQ specific to this SBO, proposers should also review the SBIR/STTR General FAQ list at: http://www.darpa.mil/work-with-us/opportunities?tFilter=&oFilter=29934. Under the HR001120S0019 summary, there is a link to the general FAQ.

Technical support for the Defense SBIR/STTR Innovation Portal (DSIP) is available Monday through Friday, 9:00 a.m. – 5:00 p.m. ET. Requests for technical support must be emailed to DoDSBIRSupport@reisystems.com with a copy to HR001120S0019@darpa.mil.