SUNS

SUNS 2023 Executive FAQ

February 3, 2023

suns@sandia.gov

Unclassified Unlimited Release
SAND2023-1160300
https://suns.sandia.gov/

Preamble

Our goal is to provide simple answers to what can often be very complicated questions. Some questions are not easy to answer succinctly, so our apologies in advance for fairly lengthy answers on key points. We strive to inform and use simple vocabulary. The answers you see reflect the opinions of the organizers of the conference – it is hoped that the workshop itself will help uncover better answers as more voices are heard.

SUNS Executive FAQ

1. What is "Software Understanding" and why is it important?

Software understanding is the practice of discovering the behavior of software by analyzing the software artifact itself, rather than relying on documentation, developer attestation, or development processes. It is important because so many vital national security and critical infrastructure functions of society rely on software, yet the way software is built today makes it inscrutable – it is difficult today to know what types of behaviors could occur when the software is used. Unexpected behavior in the software, whether intentional or not, can cause the system which depends on that software to fail or to be vulnerable to a hostile cyber actor; this in turn may cause the government mission relying on that system to fail. By using software that we cannot easily understand, in nearly every facet of government and society, we have accepted unbounded risk.

2. What is the purpose of this workshop?

Recent years have seen the integration of software into nearly every critical infrastructure and national security mission. Much of this software is not written by the government, but by third parties, and is therefore largely opaque to the government missions that depend on it. SolarWinds and other incidents have demonstrated that, in many cases, we lack adequate capabilities to understand the potential behavior of the software upon which we so heavily rely. The purpose of this workshop is to bring together a group of USG researchers with strong experience with semantic software analysis to discuss whether revolutionary software understanding capabilities might be possible which could address the nation's needs in software analysis. Workshop discussions will range over the need, advisability, and the facets of a potential national agenda and needed national software analysis community to support it.

3. What is the scope of this workshop?

The workshop focus is on potential software understanding solutions which could be developed over the next 10 years to answer a wide range of mission questions about 3rd-party software. All types of software artifacts in widespread use in national security and critical infrastructure are in scope, from executables to source code and everything in between. All types of systems and software in widespread use in these areas are in scope, including desktop, server, mobile, embedded computing, internet of things, industrial control systems, security systems, building automation, manufacturing, energy production or distribution, communication, vehicles, aviation, and many more. All layers of software in these systems are in scope, whether at the application, operating system, firmware, or other layers. Furthermore, the scope of this workshop isn't focused on one single mission question but rather on the broad range of questions that various government missions need to ask about software (see Question #7).

This is an enormous space, but it is the reality of the challenge. This enormity speaks to the kind of approaches that must be taken in order to achieve an acceptable return on investment (ROI). Fortunately, it is not necessary to provide solutions for all of the systems listed above in order to achieve an acceptable ROI. Technical strategies on how to maximize this ROI will be one of the things discussed by the technical SMEs during the close sessions.

4. What are the expected outcomes of the workshop?

Sandia National Labs has been asked by DHS/S&T to produce a workshop report, summarizing the discussions and recommendations. Among those recommendations will be a set of near-term R&D priorities which align with the workshop recommendations and can be used to inform research groups interested in improving their software understanding capabilities.

5. Aren't some of these problems (particularly supply chain) amenable to policy changes that can help? Are those in scope for this workshop?

We agree that policy changes in a variety of areas may be possible which would enable us to improve the nation's ability to answer mission questions about software. This workshop is structured around technical conversations involving a particular type of expertise – these technical SMEs are not policy experts and therefore policy is not a central focus of this workshop. However, if the technical discussions identify significant potential capability advances that depend upon certain policy changes, we will capture those in the workshop report to inform and spur further conversations with policy makers.

6. Is this workshop about software assurance?

Not directly, but there may be interest in leveraging the results of this workshop for software assurance questions, subject to certain limitations. In its strongest form, software assurance seeks to provide strong guarantees that software has no undesirable behaviors, including vulnerabilities, that could put the mission using the software at risk. Unless a particular software package was designed to be readily analyzable, software understanding techniques are extremely limited in their ability to prove that certain behaviors are absent; instead, they seek to identify what behavior may be present, but typically cannot guarantee that they have found all behaviors that are present.

However, mission owners of today routinely must make decisions about software that should be highly assured, but for which that assurance is not practical or possible to achieve today. Typically today these decisions are not informed by a body of technical evidence derived from an analysis of the software itself about its possible behavior. In this environment, software understanding would be a material advancement over the approaches already being taken but should be understood by mission owners as offering substantively weaker answers than high assurance approaches such as formal methods.

7. What do you mean by “mission questions”? Can you give some examples?

A mission question is a high-level but specific technical question that a mission owner may pose about software upon which their mission relies. These are top-down questions driven by the mission concerns. Here are some examples:

- Could my control system software permit unauthorized control of this particular critical function?
- Is there remote administration access? Is there an authentication back door?
- Does this software permit a remote user to disable this particular critical functionality of the system (e.g., a remote kill switch)?
- Does the software “phone home”? Under what conditions? What systems might it connect to?
- Could this software write to any files? Which directories might it write these files to? Which files could this program read?
- Is there behavior tied to specified dates or times?
- Is there behavior tied to specific geographic locations? (Important corollary: Does my guidance software behave differently in different regions of the world?)
- Could any local files be sent out over the network?
- Could this software record the user’s keystrokes?
- Could this software take screen shots?
- Under what conditions can the camera/microphone be turned on?
- Is the integrity of my critical data preserved by the software?
- Are there inputs that crash this program?
- What network protocols does this software handle?
- Is there encrypted communication? Is there a hardcoded key? Are the certificates checked properly? What certification authorities are trusted?
- Is there communication that is unencrypted?
- What input sensors are used to determine the action of this actuator?

A few mission questions that are asked today:

- Is there an input that will crash this program? (Fuzzing tools apply here.)
- Are there things in this software that I have seen before and discovered were bad? (Antivirus software applies here.)
- Does the program do what I expect it to do in this specific environment with this particular input? (Acceptance testing applied here.)
- Has the program been modified since it was signed? (Hashing and signature checks apply here.)

SUNS Executive FAQ

8. Where does the SBOM fit within Software Understanding?

A Software Bill of Materials (SBOM) is a list of the software components that make up a particular software application. While an SBOM can be useful in understanding what some of the functionality in a particular software application may be, there remain many questions about possible software behavior that the SBOM alone cannot answer. In effect, one particular Mission Question may be "Is this SBOM correct?" Another may be "What is the SBOM for this software?" For examples of Mission Questions that go beyond the SBOM, see FAQ #7, above.

9. Why are the sessions on Tuesday-Thursday closed? I'd love to listen in…

The main purpose of this workshop is to answer questions about what may be technically possible in the area of software analysis for understanding. There are aspects of this that are impossible, other aspects that incredibly challenging, and few aspects that are easy. The conversations will be not only highly technical and nuanced, but also easily misunderstood by non-experts. In order to cover the ground we need to, technical SMEs will need the freedom to be candid without concern of being misunderstood by mission stakeholders, policy makers, and others. Therefore, it is vital that the discussions on Tuesday through Thursday are by invitation only to select SMEs with the appropriate expertise.