The Case for an Open and Evolving Software Assurance Framework

Miron Livny[1], **Barton P. Miller**[2], **and Von Welch**[3]
[1]Morgridge Institute for Research, Madison, WI, U.S.A
[2]Department of Computer Science, University of Wisconsin-Madison, Madison, WI, U.S.A.
[3]Center for Applied Cybersecurity Research, Indiana University, Bloomington, IN, U.S.A.

Abstract – *While software is becoming an increasingly ubiquitous part of our lives, our ability to determine assurances about the resilience of that software with regards to malicious actors lags. The DHS-funded Software Assurance Marketplace (SWAMP) is an instantiation of a framework for software assurance that seeks to address this gap. The vision for the framework is not a software assurance tool in itself, but rather a mechanism to allow for flexible application of different software assurance tools and technologies into automated workflows. This paper describes the desired attributes of such a framework: tools and software packages can be easily added; analysis results, at all levels, can interpreted by different tools; and the entire software assessment process can be readily studied. A key goal of the framework is to provide the foundation for an increasingly large and productive community working on software assurance.*

Keywords: Software assurance, software development, quality assurance.

1 Introduction

Software has become an essential component of every element of our life - from the pacemaker to the national power grid. It has been growing in complexity and size at a rate that exceeds our ability to keep pace with assuring its quality. Recent events, such as Heartbleed, have exposed vulnerabilities in critical and widely used software components, software assurance (SwA) methodologies and technologies failed to prevent or detect these critical weaknesses [1]. The authors are leading a project, the Software Assurance Marketplace (SWAMP) [2], funded by DHS, to tackle the growing gap between the role of software and our ability to provide assurance about software. The SWAMP is not a SwA technology in itself, but a rather a materialization of a framework for SwA, bringing together software packages and SwA tools with the principle of continuous assurance, and forming the foundation for SwA research, development and application. In this paper we describe the framework that governs the design and implementation of the SWAMP facility, and the rationale behind it.

2 The Open and Evolving Framework

Evaluating the assurance of software involves solving a broad spectrum of problems. Each of these problems constitutes a stage in an end-to-end workflow, where each stage is realized by a collection of tools that addresses the specific tasks of that stage. Extending the impact and expanding the reach of software assurance technologies requires a model that captures the different stages in these workflows and a framework that embodies it. Such a framework will cover the stages of code development, analysis, result normalization and labeling, result merging and integration, visualization, result evaluation and annotation, and risk assessment. The model and framework cannot be static, as experience and innovation will evolve them well past the limits of today's technologies. Our understanding of software engineering and assurance challenges as well as novel methodologies will continue to grow through research and experience.

One of the key benefits of a flexible and adaptive framework is the ability to compose a variety of tools to produce a workflow that is tailored to the specific needs of a software assurance task. Operating on a chain of well-defined intermediate results, these "best of breed" tools will join forces to deliver effective assurance capabilities to the end user ranging from a student in a class to a software supply chain specialist. The value and power of such frameworks have been effectively demonstrated in a variety of research and engineering areas as they encourage and facilitate sharing within and across organizations. Easy authoring, exchange and adaptation of workflows facilitate the development and adoption of best practices throughout the community.

Another key impact of such a framework is the ease in which new technologies are adopted. By offering the "glue" needed to incorporate a new technology in a SwA workflow, the framework expedites the "the time to impact" of novel technologies. It minimizes the burden placed on the technology developers who, in most cases, do not have the means to develop the required utilities needed to make their technologies accessible to end users.

An open SwA framework will allow a developer to choose the tools and technologies that best fit their needs, and compose them into complete automated workflows. Developers benefit from the variety of tools available at each stage and the experiences of other developers, as embodied in the stored workflows, who have used the framework to solve similar problems. Developers can also bring their own tools to the framework to experiment with new technologies and methodologies. As developers gain more experience with this global view of the SwA process, they can add more tools to their workflow, and expand their coverage of the problem space. SwA Researchers and tool developers can evaluate capabilities of different methodologies and technologies.

In recent years we have seen several organizations take steps to support a more flexible and composable approach to the software assurance process. These groups include Secure Decisions's CodeDX and Denim Group's ThreadFix tools, which merge and visualize results from multiple code analysis tools, and KDM Analytic's TOIF, which serves to merge analysis tool results and represent it in a common format. Rather than presenting the end user with one monolithic software system that covers multiple stages of the SwA process, the capabilities offered by these groups allow the integration and manipulation of results from different analyzers.

3 Framework Attributes

To meet the diverse and ever changing needs and expectations of the different groups that compose the SwA community, the framework will have to offer the following key elements:

- *An environment where new tools can be added easily and efficiently:* Tool developers and researchers should be able to bring their tools into the framework with no more difficulty than bringing the tool up on their own desktop. This means not only having simply uploading procedures, but also being able to work interactively to address porting issues. Ease of bringing tools into the framework also means that once a tool is available, the operation of running it against a software package should be fully automated. Such automation requires that the framework provide the glue to run the tool against software packages with complex and even non-standard build procedures.
- *An environment where new software packages can be added easily:* As with the case for tools, software package developers should be able to bring their software into the framework with no more difficulty than bringing the package up on their own desktop. Again, interactive access is critical to keep this process simple and familiar. Once a package is successfully built, the effort of the package developer should be done. The framework must provide the glue that automates running selected tools against the package, assessing the package exactly as it would be built, and handling complex directory structures, separate compilation, whole program analysis, and builds that produce multiple executables.
- *Support for tools that integrate and interpret the output of SwA assurance tools:* The step of automating importing tools and software packages, and running the tools against the packages, are only the first steps in the workflow. While running against multiple SwA tools provides a rich source of assessment information, this information must be unified, labeled and presented to the user in a way that allows them to understand it. The SwA framework must provide open access for tools that fill all or parts of this space.
- *Access to software products and results at all levels:* An SwA framework will include analysis products from each step of the workflow. These products include the raw results from SwA tools, normalized raw results in a uniform format, merged and interpreted and labeled results, annotated results that include feedback from the programmer or higher level tools. Tool developers and programmers must have the opportunity to access any of these results and share these results, while not being forced to depend on any of them. Common data representations are key to allow the choice of multiple tools at any stage of the SwA process, and to allow independently developed tools to interact with each other.

A foundation for understanding the process of software assessment: The body of data that will be created by an active and productive SwA framework provides a life history of the software development and assurance process. As such, this data offers raw materials for study to the researcher in software engineering, software assurance, risk management, and software business processes. For example, a researcher might be studying the productivity of a software assessment method or the effectiveness of various tools and techniques. Data can be provided to researchers in both raw and anonymized forms.

4 Foundation for an SwA Community

Despite the power a framework may eventually bring, it is not obvious that any technology in itself will bridge the growing gap between software and SwA. Hence it must at least contribute to a growing community of software developers and SwA researchers working to enable SwA through education and better software development practices. Two thrusts underway are the use of the SWAMP facility in education to teach SwA earlier, alongside software development, and the development of a body of software development practices, such as structuring software in such a way to be assurable. One lesson with the recent experience with Heartbleed was that software of sufficient complexity cannot be successfully analyzed to the point it can be assured. Rather than expecting the SwA tools to bring that gap, we may need the software to meet assurance half way.

5 Current Status

The need for an open and flexible SwA framework has guided the design and development of the Department of Homeland Security Science and Technology Directorate's recent initiative, the Software Assurance Marketplace facility. As a technology-neutral entity, the SWAMP is uniquely positioned to define, implement and evolve such a framework and to make it available to the SwA community. The SWAMP is currently operational with fundamental portions of the framework, primarily for software developers operational. We welcome comments and recommendations on the framework that will help us reach and extend its impact.

6 Acknowledgments

We acknowledge the funding of the Department of Homeland Security, Science and Technology Directorate in funding the SWAMP.

7 References

[1] Kupsch, James A., and Miller, Barton P. "Why Do Software Assurance Tools Have Problems Finding Bugs Like Heartbleed?" Continuous Software Assurance Marketplace, 22 Apr. 2014. https://continuousassurance.org/swamp/SWAMP-WP003-Heartbleed.pdf

[2] "SWAMP Capabilities." Continuous Software Assurance Marketplace, 12 Dec. 2013. https://continuousassurance.org/swamp/SWAMP-WP001-Capabilities.pdf