In Support of Secure Software Education and Training

Position Statement in Response to *White House Executive Order* of May 12, 2021

Barton P. Miller
Vilas Distinguished Achievement Professor
Sohi Professor in Computer Sciences
University of Wisconsin-Madison
bart@cs.wisc.edu

Elisa R. Heymann
Senior Scientist
University of Wisconsin-Madison
elisa@cs.wisc.edu

This position statement addresses the area of the Secure Software Development Lifecycle. In particular, we address the issue of enhancing the security of the software supply chain by education and training[1].

The security of our cyberinfrastructure is only as strong as the software that we develop and deploy. As both providers and consumers of software in our supply chain, we need to ensure that the software meets the highest standard of secure design, coding, and testing practices. A key to meeting these standards is having ubiquitous software security courses in our colleges and universities and training available to our professional practitioners. The key to having such widespread teaching and training is to have open and free resources to enable such activities. As such, we advocate programs to support the development of curriculum materials available for online and in-person modes of teaching. In addition, we advocate certification standards to allow software professionals and new graduates to demonstrate their proficiencies in these areas.

The skills to design, develop, and test secure software are not widely held in the software development community. Most software curriculums concentrate on teaching the theoretical and practical skills needs to develop functional and efficient software. Where computer security is taught in the university, it usually a course that covers a broad set of topics, not a systematic introduction to security in the software development lifecycle. The software development community needs the resources to master the security issues at each stage of the software development lifecycle.

Our approach to this topic comes from more than 15 years of experience performing person in-depth software reviews to find critical vulnerabilities. This experience in vulnerability assessment has spanned the areas of scientific infrastructure to commodity software like web browsers to critical transportation infrastructure like maritime shipping. In these assessment efforts, we have found many serious vulnerabilities in the software. From those assessment efforts, we identified common design and programming practices that allowed such vulnerabilities to be present.

We then started by developing informal training materials to share with the developers of the software that we had assessed. This effort was necessary as there was little instructional material on software security readily available. As our materials started to grow in scope, we developed a pedagogy that introduced security from the first steps of a design, through the coding, and into testing and assessment. The curriculum was intended for software developers, managers, and cybersecurity professionals.

While the number of software practitioners is growing at a startling rate, there are relatively few practitioners with such software security skills. As software in systems and devices is controlling an

[1] **Executive Order** paragraph 4(e)(ix)

increasing amount of our lives, the needs for these software security skills is becoming more urgent. It is not an exaggeration to say that is has reached a point of criticality.

In response to this situation, we believe that the software community will benefit from resources that satisfy several criteria:

1. These materials should be based on a **solid conceptual framework** and not just a bag of tricks. The goal is to teach a thought process that extends beyond a particular programming language or currently understood set of threats.
2. The curriculum must be **comprehensive**, starting at the design stage, before a single line of code is written, and then carrying through the coding stage and finally the testing and assessment stages. **Security must be present at each step of the software development lifecycle.**
3. The materials should be **modular**, so that a professional programmer can quickly train on their most immediately needed skills and instructors in computer science classes can use these resources to introduce security to their topic. For example, an instructor in a database course could use the module on SQL Injection Attacks.
4. The training materials should be designed in such a way as to be **accessible to the broadest community possible**. While some modules may assume advanced software skills, there should be much available that benefits even the beginning programmer. In addition, any **video materials should be captioned**, preferably in multiple languages.
5. **Free and open access** to these materials will reduce barriers to acceptance. To support small companies and universities, cost should not be a cause for not having such materials available.
6. **Continuous development** of the teaching materials is essential to keeping them up to date with the rapid changes in the software world and in cybersecurity practices and threats.
7. An **evaluation and certification process** will allow students and trainees to demonstrate their mastery of all or parts of this curriculum. Both students and existing employees will be motivated to acquire such certification to enhance their career trajectories. Software producers will be able to advertise that they have trained teams in software security and acquirers of software will had an additional tool to evaluate the providers of their software.

We recommend a team that spans government, academia, and industry to advise such an effort. To execute the keys tasks in developing these materials, we recommend an academia-focused team. Sustained funding for such an effort is essential to keep it relevant.

We offer our current text, video, and hands-on exercise materials as a starting point for this effort[2]. These materials form the foundation for our recently developed undergraduate class, Introduction to Software Security (CS542)[3]

[2] https://research.cs.wisc.edu/mist/SoftwareSecurityCourse/

[3] https://pages.cs.wisc.edu/~bart/cs542.html