SSD: Secure Software Development

SECURWARE 2017, The Eleventh International Conference on Emerging Security Information, Systems and Technologies
http://www.iaria.org/conferences2017/SECURWARE17.html

Aspen Olmsted
College of Charleston
Department of Computer Science, Charleston, SC 29401
e-mail: olmsteda@cofc.edu

Abstract— **Secure software development is a process which integrates people and practices to ensure application Confidentiality, Integrity, Availability, Non-Repudiation, and Authentication (CIANA). Secure software is the result of a security- aware software development process in which CIANA is established when an application is first developed. Current secure software development lifecycles are simply old software development lifecycles with security training prepended to the traditional development steps and an incident response process appended to the lifecycle. Users of software applications need to have guarantees that both their data and their computing environment are not exposed to vulnerabilities during the installation and execution of the software application.**

Keywords-Cyber-security; Software Engineering; SDLC

I. Introduction

The goal of the SECURWARE 2017 special session on secure software development is to bring together researchers in cyber-security and software engineering to share and hypothesize solutions to refactoring our software engineering processes.

Current secure software development lifecycles (SSDLC) are just old software development lifecycles (SDLC) with a security training prepended before the traditional development steps and an incident response process append to the end of the lifecycle [1]. To protect the users of our applications and the data created, we need to develop the models, tools, architectures, and algorithms that support CIANA on the first day of a development project. We believe this requires revolutionary changes to the traditional SDLC to ensure security from the onset of a software development project.

For this special session, we focused on the following topics:

- Guaranteeing Data Integrity – We want to explore algorithms and architectures that will guarantee that the data an application generates will be correct.
- Ensuring Object Code Integrity – We want to explore architectures that will guarantee that the machine code of an application has not been modified.
- Secure Inter-Process Communication – We want to explore algorithms and architectures that will guarantee privacy for inter-process communications.
- Process Authentication – We want to explore architectures what can guarantee that a process is truly the software process we want to authorize and to ensure no other process can impersonate that process.
- Guaranteed Service Redundancy – We want to explore architectures that will provide service redundancy in the event of a node failure or network partition.
- Guaranteed Data Redundancy– We want to explore algorithms and architectures that will provide data redundancy in the event of a node failure or network partition.
- Secure Software Modeling – We want to explore techniques that will allow business analysts to model vulnerabilities in the early stages of the SDLC.
- Non-functional Modeling = We want to explore methods to model the properties that need to be guaranteed while the software is active.
- Software Patching – We want to explore methods that allow discovered bugs to be corrected quickly while not creating new vulnerabilities.
- Domain Specific SSD Problems – We want to find a method to share and document application domain specific vulnerabilities and solutions to close those vulnerabilities.

We do not anticipate that the topics explored in the special session will radically alter the SDLC. Instead, we hope that the exploration of new algorithms, architectures and methods can both be a first step in identifying the current gaps in secure software development and provide a tool chest for software architects to build from.

II. Submissions

The first paper in the special session is titled, Security Vulnerabilities in Javascript Hotpatching in iOS with a Commerical and Open-Source Tool and was submitted by Sarah Ford from the College of Charleston [1]. Her paper focuses on the development of mobile applications and the need for developers to be able to update applications immediately on discovery of a critical bug. She investigates the Apple iOS software patching system the limitation in their application patching lifecycle. She documents the two tools that have been developed to solve the problem and the vulnerabilities of both tools. She shoos that the tools enable quick updates but also expose users to multiple security vulnerabilities. She concludes by arguing that Apple should not allow the tools and should propose a better solution using the same technology that preserves security.

The second paper, Secure Development of Healthcare Medical Billing Software, was submitted by Paige Peck from the College of Charleston [2]. In his paper, he addresses the applied problem of Healthcare medical billing and guarantees of correctness. He argues that software in the Healthcare billing domain has been progressing into the digital era for several years, but it has been a slow and expensive process that has left many parts of the industry behind. He shows how an area overlooked in the software development improvement is security. This importance of this security is increased now that medical records are worth far more than credit card numbers on the black market. He shows evidence of hospital practices where doctors are using printed out spreadsheets to find rules for coverage of a procedure based on any insurance company's policies. He provides a solution using business rules engines and rule validations; we make it easier for a doctor or office to type in lab results and see whether a procedure will be covered by a patient's insurance company.

The final paper in the special session is Secure Software Development – Models, Tools, Architectures, and Algorithms, submitted by Aspen Olmsted from the College of Charleston [3]. In his work, Aspen documents several issues and solutions developed by students in his lab over the last five years. Solutions are developed for data integrity guarantees for distributed systems, service, and data redundancy. The solutions provided are not commercial ready products but provide evidence and a foundation for commercial grade solutions to be developed. The work also addresses earlier stages in the SDLC with solutions for modeling non-functional requirements and vulnerable application partitions. The modeling solutions leverage current industry standard modeling tools, but the standards bodies have greatly ignored the non-function security requirements in their modeling solutions. The hope is that this work will push those standard bodies to address the non-functional requirements in future iterations of the modeling standards.

III. Conclusion

Although the Secure Software Development special session at Securware 2017, touched on many important areas of the SDLC, there is still a great deal of work to do to ensure that the users of our software and the data created by our software are protected from loss and malicious users. There was no radical proposal submitted that would greatly alter the SDLC in the future. We believe the current computer science academic culture does not lend itself to radical proposal that are truly needed in this area. We hope that new risk takers are found that can help in this area. We look forward to new invocations that will be presented at sessions in future conferences of SECURWARE series.

References

[1] Microsoft, Inc., "What is the Security Development Lifecycle ?," 2017. [Online]. Available: https://www.microsoft.com/en-us/sdl/. [Accessed: 26 March 2017].

[2] S. Ford and A. Olmsted, "Security Vulnerabilities in Javascript Hotpatching in iOS with a Commerical and Open-Source Tool," in *Proceedings of Proceedings of The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, Rome, Italy, 2017, SECURWARE 2017, www.thinkmind.org.

[3] P. Peck and A. Olmsted, "Secure Development of Healthcare Medical Billing Software," in *Proceedings of The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, Rome, Italy, 2017, SECURWARE 2017, www.thinkmind.org.

[4] A. Olmsted, "Secure Software Development – Models, Tools, Architectures and Algorithms," in *Proceedings of The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, Rome, Italy, 2017, SECURWARE 2017, www.thinkmind.org.