Automating Detection and Analysis of Software Vulnerabilities in Secure Software Mitigation Methodologies

Sean Carroll

Villanova University

"This paper or presentation is my own work. Any assistance I received in its preparation is acknowledged within the paper or presentation, in accordance with academic practice. If I used data, ideas, words, diagrams, pictures, or other information from any source, I have cited the sources fully and completely in footnotes and bibliography entries. This includes sources that I have quoted or paraphrased. Furthermore, I certify that this paper or presentation was prepared by me specifically for this class and has not been submitted, in whole or in part, to any other class in this University or elsewhere, or used for any purpose other than satisfying the requirements of this class, except that I am allowed to submit the paper or presentation to a professional publication, peer reviewed journal, or professional conference. In adding my name following the word 'Signature', I intend that this certification will have the same authority and authenticity as a document executed with my hand-written signature.
Signature Sean Carroll"

Abstract

This paper explores the use of tools to automate the vulnerability detection and removal of software vulnerabilities as a part of secure software mitigation methodologies. The automation of risk analysis and threat modeling is a growing trend within industry as software security activities are performed in the software development lifecycle. Treat modeling is an activity of great complexity and requires significant expertise and funds for implementation along the secure software development lifecycle (SSDLC). Due to the sophistication of attacks that industry is faced with in current environments, it is of great necessity to implement automated tools and analysis in order to assist security teams in optimizing their secure software mitigation methodologies. An optimal secure software mitigation methodology seeks to reduce costs and increase efficiency in removing vulnerabilities along the development lifecycle of software. To achieve such efficiency, automation tools must be implemented to reduce false-positives and increase efficiency over time, as costs are mitigated by reducing the need for large amounts of qualified overseers.

Automating Detection and Analysis of Software Vulnerabilities in Secure Software Mitigation Methodologies

With the increased prevalence and complexity of advanced persistent threats (APTs), there creates a need for automation tools in secure software mitigation methodologies to sustain secure operation. The secure software development lifecycle must integrate threat modeling practices into each phase of the secure software design, rather than handle threat modeling practices during specific phases. In interest of optimizing the work of secure software development teams, an optimal approach to secure software mitigation technologies must consider the most optimal automation tools to handle specific environments of concern.

Literature Review

Kirda's (2017) publication presents a recent automated approach to threat detection using a software platform that is called UNVEIL. The automated software platform that Kirda discusses in his 2017 article focuses on ransomware detection for large-scale systems. By implementing such software solution in practice, Kirda is able to present a tool with an unprecedented efficacy, as well as document the background, research, and commercial integration process to readers that seek implementation of said commercial software. Kirda's research details a large-scale study with over 100,000 hosts in which the UNVEIL software was able to automate the process of ransomware detection without false-positives or evasion in detection. It appears that Kirda's discussed automation software platform is very similar to sandboxing of entire system environments. By monitoring incoming and outgoing data from the systems of concern, as well as monitoring open applications and file changes, the software proposed by Kirda's research has proven efficacy in detecting and removing ransomware.

Collins (2012) details the challenges in test automation practices in common agile development environments that used the practice of Scrum. Collins states in his concluding remarks that "…the difference in these scenarios regarding testing activates could be avoided with some technological solutions already reported in technological literature." Collins is direct in his assertions that test automation in agile development environments is mandatory for efficiency in practice. Collins (2012) mentions that the challenges proposed with test automation are not with the efficacy of such automation, but with integration with noncompliant developers.

Frydman (2014) proposes an automated threat mitigation tool termed AutSEC, which outputs design, implementation, and verification reports—based on models that developers create during design—for its analysis. Based on such models and figures, Frydman explains in his 2014 publication that the AutSEC solution can suggest specific ways in which security designers can mitigate and prevent vulnerabilities in their environments.

Discussion

Despite Kirda's proposed automation software platform for the process of malware detection is not novel in procedure, the one-hundred percent detection of all ransomware viruses mention in this 2017 publication is particularly new to the security detection sector. Kirda (2017) highlights that the UNVEIL software—when conducting the large-scale study—discovered threats that had never been reported by any other detection tool. Kirda in his 2017 publication details one example where a common automated malware detection tool called Cuckoo Sandbox did not detect ransomware in some test cases, while the UNVEIL software was able to detect high amounts of unusual activity taking place. Frydman (2014) proposes a novel approach to automation tool assimilation with security design teams, in which the automation tools proceeds in detection and removal of threats without disruption, but allows staggered security touchpoints

in which qualified overseers can operate and act according based on such automated analysis. Frydman's discussed approach to automation tool integration may also propose a solution to the challenges outlined in Collin's agile development teams in which conflict appeared between testing automation and the practices of secure developers.

Conclusion

Developing a secure software mitigation methodology requires qualified individuals to select solutions that optimize both the effectiveness of detection and removal of identified threats, but also allows the option for overseers to intervene to optimize and guide such procedures. The bias created by the limited data received by such tools cannot disregard the other parts of the network that are of importance in protecting systems and the software of concern. An automated detection and response to threats through the use of tools must provide equal opportunity for qualified secure developers to execute their operational roles. An optimal secure software mitigation methodology is achieved when chosen software for given environments is primed for vulnerability detection and removal, but furthermore, allows overseers to guide such technologies to ensure proper integration with all phases of the secure software development lifecycle.

References

Collins, E. F. (2012). Software Test Automation practices in agile development environment: An industry experience report. 2012 7th International Workshop on Automation of Software Test (AST), pp. 57-63. doi:10.1109/IWAST.2012.6228991

Frydman, M. (2014). Automating Risk Analysis of Software Design Models. The Scientific World Journal, 2014, 1–12. https://doi.org/10.1155/2014/805856

Kirda, E. (2017). UNVEIL: A large-scale, automated approach to detecting ransomware. 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), p. 1. doi:10.1109/SANER.2017.7884603