Hunter Richards

CS-405

Journal 8-2 – Portfolio Reflection

2/27/2024

Throughout the duration of CS-405, the emphasis on adopting a secure coding standard from the onset of software development projects has been a foundational principle that has profoundly impacted my approach to software engineering. The course literature and projects have consistently illustrated that security cannot be an afterthought but must be integral to the development lifecycle to effectively mitigate risks and vulnerabilities. The adoption of secure coding practices from the beginning not only reduces the potential for security breaches but also significantly lowers the cost and effort required for remediation later in the project lifecycle. This proactive approach to security, underscored by the principles of secure programming, has reshaped my coding habits, ensuring that I prioritize security considerations in every line of code I write.

The evaluation and assessment of risk, alongside the cost-benefit analysis of mitigation strategies, have been critical components of the course. Through practical exercises and the analysis of case studies, I have learned to effectively assess the potential risks associated with software vulnerabilities and to weigh those risks against the costs and benefits of implementing various mitigation strategies. This has equipped me with the skills necessary to make informed decisions regarding which security measures to implement, based on a thorough understanding of their impact on both security posture and resource allocation. The competency projects, particularly the development of a security policy based on coding activities, have allowed me to apply these concepts in a practical context, demonstrating the importance of a balanced approach to security risk management.

The concept of Zero Trust, a security model that assumes all users and devices, both inside and outside the organization's network, are potentially hostile and must be verified before access is granted, has been another key focus of this course. This paradigm shift from traditional security models, which operated under the assumption that everything inside the network could be trusted, to a model where trust is never assumed and must always be verified, has profound implications for how we design and implement secure systems. Through readings and project work, I have come to appreciate the Zero Trust model's relevance in today's increasingly sophisticated cyber threat landscape, where breaches can originate from both external and internal sources.

Finally, the implementation and recommendation of security policies, a significant component of our coursework, have taught me the importance of not only developing secure code but also creating and enforcing comprehensive security policies that govern all aspects of software development and operation. The projects required us to produce and present security policies framed by threat-level evaluations and to consider the role of automation in preventing coding vulnerabilities. This has not only enhanced my technical skills in writing secure code but also developed my competencies in strategic thinking and policy formulation, ensuring that I am well-equipped to contribute to the development of secure software systems that are resilient against a wide array of cyber threats. This holistic approach to secure coding practices, encompassing both technical and policy aspects, has been instrumental in preparing me for the challenges of ensuring software security in a rapidly evolving digital world.-defined security policy that is informed by industry-leading research and standards​​.

Work Cited

OWASP Foundation. (n.d.). OWASP Secure Coding Practices - Quick Reference Guide. OWASP. Retrieved from https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/01-introduction/05-introduction