Module Eight Journal: Portfolio Reflection

Throughout this course, I’ve learned that security is an essential part of the software development lifecycle and should be considered at all stages, from planning to maintenance (Federal Trade Commission, 2023). Implementing secure coding practices early ensures that each layer of defense can be tested individually and incrementally, while also avoiding major restructuring of a codebase midway through a project. Having security engineers on-site at each step can also help to detect vulnerabilities that would have otherwise gone unnoticed (Ryther, 2022).

Computer security threats are always inevitable, but I’ve also learned about the ways potential risk can be categorized using a threat matrix (Irwin, 2023). This can be helpful for putting more focus into detecting larger, priority threats quickly and mitigating the breach of important assets. The quicker a threat is detected, the lower the remediation cost (SecurityScorecard, 2024), and in some cases these costs can be entirely averted if extensive monitoring is implemented. The increased complexity and cost of adding extra defense layers is one downside, but I’ve learned that it’s important to weigh this tradeoff with the importance of the assets being protected.

Later in the course, learning about the concept of “zero trust” reinforced some of the earlier principles for me - such as “default deny” and “least privilege”. It makes sense to maintain authentication protocols with a zero-assumption approach; any time this process is skipped, it gives a potential opening for an attacker. Originally, I wasn’t sure if it was best to implement zero trust in *all* cases considering the trade-off in user convenience, but I think that in today’s world it should be required for any organization which needs to protect vital company or user data. Mentioned in my last discussion post, a middle-ground is used by most login servers today which remembers user logins locally, while still requiring passwords and prompting for multi-factor authentication (MFA) on any new device.

Lastly, implementing and presenting the security policy throughout the milestones and up to the end of Project Two was also insightful; by learning about concepts such as Defense-in-Depth (DiD), coding standards, Triple-A (authentication, authorization, and accounting), various types of encryption, and the DevSecOps pipeline, I was able to consider the current policies critically and bring up my own suggestions for improvement. I hope to use this knowledge for future educational and career endeavors!

References

1. Federal Trade Commission. (2023, August). *Start with Security: A Guide for Business.* ftc.gov. <https://www.ftc.gov/business-guidance/resources/start-security-guide-business>
2. Ryther, J. (2022, February 6). *Don’t Leave Security to the End.* Medium. <https://medium.com/@jryther91/dont-leave-security-to-the-end-ca13b107d4>
3. Irwin, L., (2023, May 10). *What Is a Cyber Security Risk Assessment Matrix?* Vigilant Software. <https://www.vigilantsoftware.co.uk/blog/what-is-a-cyber-security-risk-assessment-matrix>
4. SecurityScorecard. (2024, March 5). *Remediation vs Mitigation in Cybersecurity: Understanding the distinctions and strategic applications.* SecurityScorecard Blog. <https://securityscorecard.com/blog/remediation-vs-mitigation-in-cybersecurity-understanding-the-distinctions-and-strategic-applications/>