Mitchell Ibarra

Southern New Hampshire University

CS 405 Secure Coding

8-2 Journal: Portfolio Reflection

4/25/2021

*Adoption of a secure coding standard, and not leaving security to the end*

Adopting secure coding standards and not leaving security to the end can save a lot of future headaches and save time and money. Waiting until the end of the software development lifecycle to implement security will undoubtedly leave vulnerabilities and possible unsecure practices in the code, even if it has been code reviewed well, people are human after all and it is difficult to foresee future integration issues.

*Evaluation and assessment of risk and cost benefit of mitigation*

Prior to or while creating a security policy, risk and cost benefit analyses should be conducted to determine what areas are the highest priorities, which threats are the most likely and unlikely, and what steps can be taken to handle all these threats. Risk matrices are beneficial to creating lists like this which can help lead to which security policies should be put in place.

*Zero trust*

In *A Practical Guide to Zero-Trust Security*, Kueh describes what zero trust means and the five main pillars of zero trust: device trust, user trust, transport/sessions trust, application trust, and data trust. One of the overarching components of zero trust is the concept of leas-privilege access. This ensures that a user or system shall only have access to the least number of resources needed to complete their task (Kueh, 2020). Implementing this policy within a security policy inherently protects against unintended data leaks or breaches by design. One such case where this would have been beneficial is the Equifax breach in 2017.

*Implementation and recommendations of security policies*

One such implementation of security policies is to incorporate a DevSecOps pipeline as the Software Development Lifecycle. The goal of DevSecOps is to inherently make security part of the software development workflow with secure coding best practices and testing automation so that it is attempted to be bolted on later in the cycle (Vijayan, 2019). Other recommendations of security policies is to incorporate threat modeling and training on secure coding. Then developers will be on the front lines of ensuring that code meets security standards and can get in front of removing a lot of potential risks early in the process.

References

Kueh, T. (2020, January 15). *A Practical Guide to Zero-Trust Security*. Threatpost English Global threatpostcom. <https://threatpost.com/practical-guide-zero-trust-security/151912/>.

Vijayan, J. (2019, December 31). *6 DevSecOps best practices: Automate early and often*. TechBeacon. https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often.