In my work throughout this course, I’ve learned that adopting secure coding standards from the very beginning is crucial. Security shouldn’t be an afterthought added at the end; instead, it must be woven into every phase of the development process. This means designing systems with threat modeling in mind and integrating automated tools that check for vulnerabilities as the code is being written, which not only prevents issues from compounding but also saves time and money in the long run.

Evaluating the risk and weighing the cost of mitigation measures is also essential. By continuously assessing which vulnerabilities pose the greatest threat, it becomes easier to determine the most effective and efficient ways to address them without unnecessarily burdening the development process. My experience with Defense in Depth has shown that while layered security measures are important, they need to be balanced with usability and system performance to ensure that security enhancements do not inadvertently slow down progress.

The concept of zero trust has also significantly influenced my approach to security. Unlike traditional methods that assume internal networks are safe, zero trust requires that every access request be verified, regardless of where it comes from. This continuous verification process, which covers devices, users, sessions, applications, and data, creates a more robust security environment that adapts to the dynamic nature of modern IT systems.

Finally, implementing strong security policies is key to bringing all these elements together. A good policy ensures that secure coding practices, continuous risk assessment, and zero-trust principles are not just theoretical ideals but practical, everyday practices. In my view, policies must be clear and adaptable, promoting a culture where every team member is responsible for security. This comprehensive approach not only enhances the overall security posture but also supports a more efficient and innovative development process.

**References:**

Vijayan, J. (2020, March 9). *6 DevSecOps best practices: Automate early and often*. TechBeacon. Retrieved March 1, 2025, from [https://web.archive.org/web/20240117143629/https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often](https://web.archive.org/web/20240117143629/https:/techbeacon.com/security/6-devsecops-best-practices-automate-early-often)

Kueh, T. (2020, January 15). *A practical guide to zero-trust security*. Threatpost. Retrieved March 1, 2025, from <https://threatpost.com/practical-guide-zero-trust-security/151912/>