Tyler Pimental

CS-405

Professor Alesso

03/02/2025

8-2 Journal: Portfolio Reflection

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

Throughout this course we have explored DevSecOps practices. As written by Vijiayan, “Security controls and tests need to be embedded early and everywhere in the development lifecycle, and they need to happen in an automated fashion...”(Vijayan, 2022). He speaks to that theory here that you are essentially only contributing to your tech debt, by waiting until the end of an applications core development stage to consider security. Developing security features can have significant impacts on performance and architecture that should be addressed as early as possible, as to not have a cascading effect on features to come.

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

I believe the largest risk an organization can have when it’s related to security, are ransomware attacks. These can be nasty scenarios that can either lead to severe financial or data loss. “It’s important to understand that application security is as critical as email security in defending against ransomware and other malware... Vulnerabilities in a public-facing internet MSP application were exploited to spread ransomware to their customers.” (How Hackers Exploit, 2023) While ransomware attacks can be more frequent to target individuals, it is still possible for bad actors to leverage poorly secured application code, to find access points into systems.

**Zero trust**

Zero trust is a great principle that forces defense in depth on a smaller scale. Even if you believe all of your resources and systems are secured behind a DMZ, you should still be treating these connections as untrusted. This could mean, designing your application to only accept secure transaction or signed information from microservices.

**Implementation and recommendations of security policies**

When it comes down to the actual implementation of security policies, especially in a development environment, there are areas that could be taken less seriously than others. For example, if certain security policies only allow for FIPS compliant cryptographic methods, however, to ensure you have maximum computability with your development tools, you may increase the allowed cryptographic methods to some less secure ones. Even though you might make this change in a non-public facing development environment, the security risk you open is still real. The recommendation and implementation of security policies should be taken as serious as possible, no matter the location or environment.

References

BrendaCarter. (2025, February 27). *What is Zero Trust?* Microsoft Learn. <https://learn.microsoft.com/en-us/security/zero-trust/zero-trust-overview>

*How hackers exploit poor application security in ransomware attacks*. (2023, March 17). Barrcuda Blog. <https://blog.barracuda.com/2021/07/08/how-hackers-exploit-poor-application-security-in-ransomware-attacks?form=MG0AV3>

Vijayan, J. (2022, November 16). *6 DevSecOps best practices: Automate early and often*. TechBeacon. <https://web.archive.org/web/20240117143629/https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often>