8-2 Journal: Portfolio Reflection

Keegan Sevener

CS-405 – Secure Coding

06/24/2025

Over this course, I have learned more about secure coding and how to implement it. The most novel standard I learned about was the adoption of secure coding standards through a security policy. A security policy is a formal set of rules that an organization creates to articulate how to conduct their security services and protect sensitive and critical information (Seacord, 2013). Security policies are very important for any software organization as they explicitly outline the standards and coding habits/behaviors that are expected of employees at an organization (Dunham, 2020). This eliminates confusion and creates better accountability amongst an organization.

Security policies eliminate ambiguity in security priorities. Implementing a security policy helps transform an organization from DevOps to DevSecOps which addresses security at every stage. Moving to DevSecOps shifts security left instead of leaving at the end. Shifting security left means addressing security concerns at the beginning of the software development lifecycle, SDLC, (*Dora: Devops capabilities: Shifting left on security*, n.d.). Organizations which shifted left on security saw the time spent on fixing security issues cut in half (*Dora: Devops capabilities: Shifting left on security*, n.d.). This is why it is important to address security from the beginning. It not only makes software safer but also easier to develop said software. If security is left until the end, then a lot of refactoring has to occur to integrate security within a program.

An important part of early security implementation is risk assessment. With risk assessment, we can model potential threats to a system in the design phase to predict what our potential threats/risks are and model security around that (Seacord, 2013). Threats come in many forms and depend on the type of system being developed. Based on the type of system being developed, we can determine what kind of vulnerabilities we could encounter in code and work to mitigate those. For example, if the program is using a Structured Query Language, SQL, database, then we will need to be mindful of SQL injections and ensure our queries are constructed using prepared statements.

We also need to keep the type of hackers that would want to break into our system in mind. Some are just looking for a quick buck, while others are interested in a system for the “KUDOS” and being able to say they breached a system (IBCShow, 2018). We also must be mindful if our system is politically affiliated in some way. Some hackers are more tenacious given they are driven by ideological reasons which means they will not be repelled by a few failed attempts (IBCShow, 2018). Knowing what type of data we are protecting and who we are protecting it from can help us not only protect data but mitigate fallout from security breaches when they occur. By employing mitigation techniques such as the principle of lest privilege or network segmentation, we can limit the scope of data and system resources that an attacker will have access to if they should gain entry.

One way to prevent security breaches and mitigate damage is to employ a zero trust system. Zero trust is the practice of authenticating every connection and thus treating both internal and external devices the same way (Brook, 2024). Zero trust means always verifying access attempts and not giving anyone a free pass. Zero trust shifts security away from a perimeter style approach to a much more granular one. This enables better scalability for an organization and better security that is more fitted for today’s environment (Brooks, 2024). Zero trust can enhance data protection and mitigate disaster from security breaches.

Creating a security policy and implementing a zero trust framework is the best way to create a DevSecOps framework. Zero trust should be the default standard for organizations. Security policies should always address the most pertinent security issues for an organization and outline them by priority. To fully and properly implement a DevSecOps framework, an organization must ensure that it is understood that security is everyone’s responsibility. This must be outlined in the security policy. Following these principles will reduce refactoring, add security more easily, and create a well-understood guide for secure coding.

References

Brook, C. (2024, October 23). *Understanding the zero trust security model to safeguard digital*

*infrastructure*. Digital Guardian. <https://www.digitalguardian.com/blog/understanding-zero-trust-security-model-safeguard-digital-infrastructure>

*Dora: Devops capabilities: Shifting left on security*. DORA | DevOps Capabilities: Shifting Left

on Security. (n.d.). <https://web.archive.org/web/20240524180526/https://dora.dev/devops-capabilities/process/shifting-left-on-security/>

Dunham, R. (2020, May 5). *The importance of a company information security policy*. Linford &

Company LLP. <https://linfordco.com/blog/information-security-policies/>

IBCShow. (2018, June 21). *What motivates hackers?* [Video]. YouTube.

<https://www.youtube.com/watch?v=BoWouwPGtVY>

Seacord, R. (2013). Secure coding in C and C++ (2nd ed.). Pearson Education, Limited. June 20,

2025, <https://mbsdirect.vitalsource.com/reader/books/9780132981972>