Cheryl Buzicky

Module 8 Journal: Portfolio Reflection

Professor Ramsey Kraya

06/29/2025

When reflecting on all of the things I learned in this course, one of the most important lessons is that security must be **integrated into the software development life cycle from the very beginning,** not left to the end as an afterthought. Implementing a **secure coding standard** early on, such as using safe exception handling practices in C++ and validating input data helps eliminate vulnerabilities before they become serious security risks (Cppreference.com). By integrating security directly into a program’s design, developers reduce both the cost and complexity of addressing issues later in production, where fixing them can be exponentially more expensive. The average data breach costs a company $4.9 million not including the damages done to an organization’s reputation (IBM, 2024). Adopting a set of secure coding standards is one technique that can be used to strengthen software, helping developers avoid common vulnerabilities, write safer code by default, and create systems that are more resilient to attacks.

Another thing to consider is the **evaluation and assessment of risk.** In today’s digital world, developers must assess not only technical vulnerabilities but also the **likelihood and impact** of various attacks. Real-world incidents, such as the LinkedIn data breach or the WannaCry worm, demonstrate how weak security policies can have devastating consequences. A well-planned risk assessment process considers both the need for various measures and the **cost of implementation.** This allows a company to make informed decisions about where to apply mitigation strategies, and how deeply to invest in layered defenses.

Development teams should also design programs around the concept of **zero trust.** In this model, it is assumed that everyone using a program has the potential to exploit it, and even the most secure networks can still harbor vulnerabilities. Marcus Hutchin’s story in Wired illustrates how even skilled insiders or security researchers can walk the line between ethical and unethical behavior, further supporting the idea that no one should be automatically trusted (Greenberg, 2020). The principle of zero trust aligns with what Cairns and Somerfield (2017) describe as: “If you don't control it, you can't trust it”. Forcing users to use multi-factor authentication, as well as monitoring user login and activity data can help detect and prevent malicious attacks, even from the most trusted sources.

To be sure software programs remain secure in today’s threat climate, security policies should include data encryption, input validation, strong authentication protocols, and a detailed incident response plan. These policies need to be part of the initial DevOps planning stages, so they are embedded into the actual product design and implementation. For example, LinkedIn’s own response to their data breach involved both technical fixes and changes to user safety policies (LinkedIn, 2016). When combined with secure coding practices, these policies can help teams implement “defense in depth,” where security is implemented in layers so that if one layer fails, the others continue to protect the system. (Yendamury & Mohankumar, 2021).

Secure software development means an early commitment to secure coding standards, continuous risk evaluation, zero trust principles, and well-implemented security policies. These components work together to build systems that are not only functional, but resilient against both current and future threats.

**References**

Cairns, C., & Somerfield, D. (2017, January 5). The Basics of Web Application Security. martinfowler.com. https://martinfowler.com/articles/web-security-basics.html

Cppreference.com. (n.d.). std::exception. Retrieved May 28, 2025, from https://en.cppreference.com/w/cpp/error/exception

GeeksforGeeks. (2024, July 29). Types of Hackers. https://www.geeksforgeeks.org/types-of-hackers/

Greenberg, A. (2020, May 12). The confessions of the hacker who saved the internet. Wired. https://www.wired.com/story/confessions-marcus-hutchins-hacker-who-saved-the-internet/

IBM. (2024). Cost of a Data Breach Report 2024. https://www.ibm.com/reports/data-breach

LinkedIn. (2016, May 18). Protecting our members. https://www.linkedin.com/blog/member/trust-and-safety/protecting-our-members

Yendamury, G., & Mohankumar, N. (2021). Defense in depth approach on AES cryptographic decryption core to enhance reliability. 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 1–7. https://doi.org/10.1109/iemtronics52119.2021.9422567