### Southern New Hampshire University

# 8 – 2 Journal: Portfolio Reflection

### Bairon Gomez

### Professor Alex Pettit

### CS 405 – Secure Coding

October 20, 2023

Adopting secure coding standards is beneficial for developer, the company, and their users. It can strengthen security against any potential attacks and data leaks by having developers abide with the coding standards of the company. Coding standards vary from company to company, but they all have the same goal in mind. The goal is to deliver a platform that users are comfortable using without worrying about data breaches that can negatively impact them. Companies establish coding standards to provide a foundation of expectations that need to be met by developers while creating code. This in return ensures not only that developers create quality work, but also delivers safe products. Benefits of delivering a system that is secure is limitless. Some of the benefits are avoiding stressful data breach path up, losing users trust, and diverting expenses to fix the damage left behind. One common standard that holds a strong impact is “not leaving security to the end”. The purpose of this practice is to incorporate security testing as soon as possible to avoid problems from escalating out of control by leaving testing until the end. Neglecting to test for any bugs in the code can lead to more issues if a developer continues to code on top of vulnerabilities. Focusing on security in the early stages of development will protect the users and the company from data leaks that can harm everyone.

One of the many ways to understand what needs to be protected and how it could be protected is by creating a strategic architecture plan on what needs to be protected and how. Understanding what is valuable in the system and how to safeguard it against a potential breach will mitigate the damage that can be done if a data breach occurs. Learning how data breaches occur is another positive aspect of understanding how to protect the system. When a developer learns how attacks are implemented they can also understand how to prevent these vulnerabilities. In return it will secure the system and train developers to be aware of vulnerabilities that can be introduced while coding.

One popular security principle that is valuable and essential to creating a safe platform is known as zero trust. This principle requires the system to be skeptical of all user inputs. In other words, it suspects that all inputs contain some sort of malicious content. This principle ensures that all inputs are validated before they are granted access into the system. The gateway to data breaches is at the entrance of the data received by users. Regardless of who the user is, it is important to validate that the input does not contain malicious malware. This can be done by authenticating connections and data input. “In a for-profit company, zero-trust can reduce cybersecurity premiums and enhance the company's overall risk profile with possible positive effects” (Maclean & Stewart, 2022). Validating users credentials before they can assess the system, validating their input, and ensuring that the user only does what they are authorized to do are all methods of zero trust principle. It can prevent and mitigate unathorized users from entering and making changes into the system.

There are many methods that can be used to implement secure coding practice in a system such as defense in depth and Triple-A. These are two methods I would recommend to secure any platform due to their security characteristics. Defense in depth uses multiple layers of security to mitigate any attack by having layers of security defend the system. If one layer of security is breached, the attacker must deal with other layers of security to access the systems data. Defense in depth is not limited to a particular set of security layers. The implementation of different layers are left up to the company or developer to decided. The layers used in defense in depth can consist of encryption, perimeter security, physical security, and more. Triple A on the other hand focuses and authorization, authentication, and accounting. This security method used together focus on monitoring the actions and accessibility of the user. Before implementing any security method it is important to ponder what should be protected in the system. Where is the information stored and how can it be secured from any attack. Once this is done, we can then implement defense in depth and triple-A to focus on the areas that need protection.

Resources

Maclean, D., & Stewart, R. (2022, September 22). *Zero-trust security: What architects need to know*. Enable Architect. https://www.redhat.com/architect/what-is-zero-trust

Murray, A. (2020, June 13). *Secure coding: A practical guide*. Mend. https://www.mend.io/blog/secure-coding/

Mylonas, L. (2018, November 27). *What is AAA Security? an introduction to authentication, Authorisation and Accounting*. Codebots. https://codebots.com/application-security/aaa-security-an-introduction-to-authentication-authorisation-accounting

Nidecki, T. A. (2023, January 18). *5 major benefits of early security testing*. Invicti. https://www.invicti.com/blog/web-security/5-major-benefits-of-early-security-testing/#:~:text=Early%20security%20testing%20means%20that%20code%20is%20never,stages%20and%20assigned%20to%20someone%20else%20to%20fix.

Watters, A. (2021, January 25). *What is defense in depth and how can you achieve it? pro tips for proactive cybersecurity*. CompTIA. https://www.comptia.org/blog/what-is-defense-in-depth