Portfolio Reflection

Brandon Cook

Southern New Hampshire University

CS 405: Secure Coding

Dr. Mimi Tam

August 22, 2025

Portfolio Reflection

Security should never be left to the end of a project, whether that is when developing software, managing projects, or dealing with system architectures. One of the greatest threats to any operation in today’s modern era is malicious actors, intentionally or not, who can cause damage to a system. Whether that is through intentionally breaking a system to get in and exfiltrate data, or by an internal employee or user who clicks on a malicious link, or any step in-between or beyond. By baking security measures such as zero trust, default deny, network segmentation, input verification, principle of lease privilege, and so on, into a project, the end result is more secure, clean software. Implementing security later in the design phase can easily ramp up time and costs as the project needs to be refactored and rebuilt to incorporate important policies.

As with any design, the specific threats and issues that the product may face should be evaluated. Questions should be asked to determine the importance of the functions of the application or system. For example, does the system need external input to function? Does personal data need to be stored? Will our system need to connect to the internet? Considerations like that lead to a set of requirements, which can then be sorted and prioritized. If personal data is being stored, that data needs to be secured with the best algorithms currently available. A data leak in that case would be catastrophic to both reputation and revenue. High priority items should include those that will either halt operations, thus halting business, and those that will lead to a lost of trust and data. Of course, security should surely be implemented in every aspect of a product.

Zero trust is a great security policy to employ regardless of the risks. All security standards should look into it as a basic option for protecting operations and systems. Assuming that anything that connects to a system, whether that is a user, device, or other connection, to be untrusted leads to better security implementation. Better policies are practiced, such as input sanitization, authorization, and auditing procedures. Explicit denial of anything that isn’t extensively parsed, filtered, and granted provisional access leads to a safer environment. While this type of policy can be difficult for users and employees to adapt to, user training on why this policy needs to be implemented, citing references of real-world cyber attacks such as the Target breach of 2013 and the PowerSchool breach of 2024, can help ease concerns.

Ideally, a well-defined security policy will lead to smoother operations as everyone is on the same page. In reality, security is a constantly growing and changing field that requires constant vigilance and improvement. A security policy should also grow to adapt to better practices and mitigation techniques against vulnerabilities and exploits, which are constantly being discovered. No security policy is 100% perfect, which is why the policy should also include an incident response plan that details how an organization will respond to an emergency, whether that is an active attack from an outside organization or an internal user accidentally sending corrupted input through a system. It needs to be understood that security does not mean that a system is protected; however, the depth of the policies in place can determine how many incidents are avoided and deterred.