Southern New Hampshire University

CS 405: Secure Coding

Portfolio Reflection

Justin Vallia

December 15, 2021

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

Adopting a secure coding standard is something that every development team should be considering strongly if they do not already have these in place. The security of any project, especially in today’s world, should always be in the forefront of the SDLC. It should be considered in the beginning, all throughout the project with unit testing and at the end prior to deployment. Security is not something that should ever be solely left to the end to deal with. As cyber criminals and would be attackers grow stronger everyday so must our secure coding standards and the ways we implement them in everything we work on. By implementing these security measures at every step along the way and in utilizing defense in depth measures we can strongly reduce the chances of vulnerabilities and attacks to our projects and products.

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

Knowing your enemy is essential in any form of security. This is no different, we must make ourselves aware of the potential threats, internal and external. Once familiar with emerging trends in security we must make a risk assessment as to how likely each scenario is or could be. This risk assessment is what should determine our decisions. The challenging part of security in this field is that the threats are ever changing and evolving at a rapid pace, therefore so must we change and modify our security measures. In the event we find ourselves becoming complacent and/or lazy then we could very well find ourselves in hot water because it is crucial that we address each and every threat and this is the only way we can truly mitigate our risks that are accompanied by each threat. The driving force we have in our favor is that we can begin implementing our security measures is by incorporating very simple secure coding techniques and standards within our security policies. One of the most important of these should be that we always remember to conduct testing as early and as often as possible. While there are many tools at our disposal we must be able to evaluate our needs on a case-by-case basis and from there utilize these tools to truly mitigate our risks.

**Zero trust**

The world we find ourselves living in is by no means perfect and as such this translates over into our field of study. In addition, the common level of authentication methods tends to be failing and due to this we must assume a level of zero trust all the time. The most effective solution to this problem is to perform verification for each individual user on each individual device. This model is commonly referred to as “Never Trust/Always Verify”. Traditionally we are all familiar with basic username and passwords however as we continue to change and grow with our surrounding we are finding that utilizing tools such as two-factor authentication and even biometrics to replace these more traditional methods. The security policy that is put in place will determine if a device and user can be trusted to gain access to our systems and projects we are working on. With the development of technology, we must ensure we are going above and beyond to use the data loss prevention technologies available as we see fit for our needs.

**Implementation and recommendations of security policies**

Long gone are the days where the creation and implementation of a security policy is a choice. Instead, this has become a necessity in order to survive in this world full of imminent risks and attacks. The security policy represents the core principles and standards which are to be followed at all times in any development project we find ourselves working on. The development of software requires the implementation of this policy to be continuously governed, implemented and maintained at all times. Other items which should be take note of are to ensure we always keep our software up to date, apply our patches to software quickly and efficiently, ensure we are performing our unit testing as early and as often as possible, and require authentication changes on a regular basis. By keeping all of these best practices in mind and in place they then add another layer of defense in depth which in turn combats and mitigates the risk of attacks.

**References:**

*Confluence*. Top 10 Secure Coding Practices - CERT Secure Coding - Confluence. (n.d.). Retrieved December 8, 2021, from https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices

*What is Zero trust security? principles of the zero trust model*. crowdstrike.com. (2021, December 2). Retrieved December 15, 2021, from https://www.crowdstrike.com/cybersecurity-101/zero-trust-security/