8-2 Journal:

Portfolio Reflection

Brandon Ricks

CS 405

Ahlam Alhweiti, M.S

Southern New Hampshire University

10/21/21

This is your final course reflection as well as your portfolio submission. Reflect on and include a discussion of the following topics, using readings from throughout the course to support your views.

Over this semester, we have covered many topics on secure coding, and compiled this information together into the security policy. In this reflection, we will go back over some important topics to reflect on.

Adopting a secure coding standard helps to mitigate possible vulnerabilities in security, and also work to remove errors that can put a system at risk (Foster, 2018). In this class, we primarily used SEI CERT coding standards, which are primarily used with C and C++, as well as Java (Foster, 2018). However, this is just one of many standards out there, and all work towards similar goals.

Not leaving security to the end is about working early and preventatively to remove risk and tighten security from the start. An insecure application, device, or program that hasn’t made use of early adoption of secure coding and standards will likely require extensive rewriting and testing (costing time and money), and if left unchecked, could cause potential issues to users (Bellairs, 2018). Not leaving security to the end helps to work preventatively, rather than reactively.

Risk assessment and cost/benefit analysis helps to frame the risks, costs, and benefits of adopting a particular practice or standard. Most coding standards have this shown in the form of a table, with threat level, severity, cost to implement, and overall importance shown, in order to help determine the priority. Many studies have also been done to compare the cost and benefit associated with secure coding, and typically the result is the benefit of adopting early security, good coding standards, and overall limiting risk, will outweigh the cost (Arora et al., 2013).

“Zero Trust” ties in with AAA framework, as well as setting the baseline by which one has access. The standard approach in most cases is to trust but verify, meaning to not default to distrusting, and to verify access. Zero trust means to not automatically trust anything, but to simply verify everything (Pratt, 2018). This attitude towards interactivity and connectivity between devices and users is very preventative, not assuming anything, whether inside or outside a network or system. It strongly incentivizes the “nothing is safe” approach towards security.

In closing, the development of the security policies and recommendations helps to bridge the gaps between each of these concepts, and more, by attempting to adopt all aspects of secure coding in some fashion. This was to transition DevOps into DevSecOps, adopting security standards and practices throughout a development cycle.

References:

Arora, A., Telang, R., & Frank, S. (2013). *Estimating Benefits from Investing in Secure Software Development | CISA*. Cisa.gov. <https://us-cert.cisa.gov/bsi/articles/knowledge/business-case-models/estimating-benefits-from-investing-in-secure-software-development>

Bellairs, R. (2018). *What Is Secure Coding? | Perforce Software*. Perforce Software. <https://www.perforce.com/blog/sca/what-secure-coding>

Foster, S. (2018). *What Are Security Standards? Secure Coding Standards Overview | Perforce Software*. Perforce Software. <https://www.perforce.com/blog/qac/secure-coding-standards>

Pratt, M. K. (2018, January 16). What is Zero Trust? A model for more effective security. CSO Online. <https://www.csoonline.com/article/3247848/what-is-zero-trust-a-model-for-more-effective-security.html#:~:text=Zero%20Trust%20is%20a%20security,to%20don’t%20trust%20anyone>.