**8-2 Journal: Portfolio Reflection**

Alex Hitchens

Southern New Hampshire University

CS 405 – Secure Coding

Prof. Olga Mill

October 26, 2025

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

Throughout the class, it was emphasized that adding secure coding standards early and often leads to a much better outcome. The first (and arguably most important) reason for doing this is for the safety of your users, ensuring their data or PII is secure, as well as protecting your own company from possible legal issues. The second reason is because the earlier you enact a coding standard, the less expensive it is to take corrective actions. Much like the foundation of a house, if your code foundation is not secure, you will have to peel back more layers to fix it. The third reason to not wait until the end is that everyone should be responsible for security. This means that it should not just be one team that goes through near the end and fixes things, it should be a team effort that is enacted by everyone who works on the process.

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

Evaluating risk and cost benefit of a project is a way of enacting a digital triage system what order problems should be addressed in. In the Navy, we called it ORM or Operational Risk Management, but at the end of the day it is the same process of taking how severe a problem would be if it happened and considering the likelihood of that problem happening. For instance, a problem that is could be “decently severe”, but very unlikely, may be less important than a problem that is only “kind of severe” but happens all the time. This information can be represented in a variety of ways from a risk matrix to a point score, depending on how your organization handles it. On my last several ships it was a color-coded point score that not only looked at severity and likelihood but also experience of the assigned personnel (new sailors were more of a risk in a position than someone who had done it many times before). However an organization chooses to implement it, this method can give you a good road map of what problems to tackle first.

**Zero trust**

Zero Trust is something that I have worked with for a long time in the Navy and is the mentality of requiring security checks be implemented both in and out of the home network. Old “castle and moat” system designs assumed that once someone had logged in and verified, they were safe to interact across the network. Time after time we have seen insider threats as well as traditional outside bad actors gain access and cause Havok within a system. How to counter act this? Zero Trust. First, adopt a AAA policy across the board. Authenticate users in the traditional way to include mandatory strong password policies, password change policies, and two factor authentication. Authorization means that you implement user roles to keep one user from having the power over the entire system. This is only part of the story, the other part is ensuring proper coding standards are being followed to prevent attack vectors from bad actors. This includes external input coming in, as well as information being sent between internal methods and functions. Lastly, the last line of defense is encryption. In flight, rest, or use, data should be encrypted to prevent exposure should a bad actor find a way in that wasn’t accounted for. While no system is 100% secure, taking on a zero-trust mindset makes your organization a hard target for hackers.

**Implementation and recommendations of security policies**

Security policies ensure everyone is on the same page. It is not enough to simply say “Write good code” you should also define what good code is. Nasa for instance has it’s “Power of 10” where they spell out 10 very important coding standards specific for their mission, mostly involving keeping code from running forever, and ensuring code simplicity be limiting the number of nested operations. Their security policies are very strict because they software must control a “butterfly on a bullet” as the shuttle missions were described. Security policies should be tailored to the organization they are made for. On top of this tailoring, another major benefit for this process is that it ensures everyone follows the *same* standard. For instance, you may may have two departments, that each have two separate code standards, that do not come into conflict with one another. The best way to solve this is to have a organization wide security policy but allow individual departments to go more detailed if necessary (what we do at my job).

**References:**

Kueh, T. (2020, January 17). *A practical guide to zero-trust security*. Threatpost English Global threatpostcom. https://threatpost.com/practical-guide-zero-trust-security/151912/

Foster, S. (2024, May 28). *NASA’s 10 rules for developing safety-critical code*. Perforce Software. <https://www.perforce.com/blog/kw/NASA-rules-for-developing-safety-critical-code>

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