Sean Bucklin

[sean.bucklin@snhu.edu](mailto:sean.bucklin@snhu.edu)

CS-405

8-2 Journal

Portfolio Reflection

We live in a world where there is a constant danger of having your data breached. This applies to personal and corporate data. Despite best efforts, it’s really to the point now where it’s not a matter of if you will have a security breach but when and what will be the impact of it. Throughout this course we learned about the different types of hackers and their motivations. We also learned of various tactics a hacker may employ to breach an application, such as exploiting a buffer overflow or taking advantage of SQL injection. Between the understanding of secure coding techniques to mitigating common software vulnerabilities and understanding the motives of hackers, it becomes very apparent that software security cannot be left until the end.

Adopting a security policy is an important step towards not leaving security until the end. A security policy can identify the aspects of security that are most import to an organization, and most applicable to the software they develop. It will help to get the entire organization thinking about security early and help to hold everyone accountable as well. There are several phases of a project in which security must be thought about and incorporated such as the design, development, testing, and maintenance phases.

Overall, a key takeaway from the work in this course is that the best security has a layered approach. As already stated, it’s really not a matter of if there will be a security breach but when and what will the impact be. This is where security best practices such as defense in depth, Triple A, and zero trust come in. The five pillars of zero trust, which are device trust, user trust, transport/session trust, application trust, and data trust shine a light on the multiple points of security that are required and how you may apply a defense in depth strategy.

Triple A defense is one strategy that really stuck with me. Authentication, authorization, and accounting is a logical three-tiered approach to securing software. Authentication deals with verifying a user is who they say they are and that they have been given permission to access an application. Authorization is a layered security approach to give users specific access to certain activities and data. Lastly, accounting is crucial for monitoring how a user has interacted with software, and what data they have accessed. Accounting is an aspect that stood out to me the most because it really helps with responding to a security a breach and understanding the impact of the breach.

Software security is a vast topic. From taking a very focused view of mitigating specific code vulnerabilities, to taking a much broader and wide sweeping view it becomes very apparent that security cannot be left to the very end. There are too many points at which security could fail. Thinking about it early and often will allow you apply security strategies such as defense in depth, zero trust, and triple A. If left until the end it would be impossible to retroactively applies these principles successfully. The software would have to be developed again from scratch at great cost to an organization.