8-2 Journal: Portfolio Reflection

Madeline McNeeley

December 16, 2022

CS-405

Incorporating security policies earlier in the software development lifecycle or shifting security left ensure that security is not being left to end of the cycle. By implementing security earlier in the process, vulnerabilities can be addressed earlier or avoided all together. One benefit to catching vulnerabilities sooner than later is it cost less and takes less time to remedy. To adopt secure coding standards, security should be taken into consideration as early as the architecture and design stages of the SDLC. During the development stage, security vulnerabilities should be identified and addressed using methods such as static testing and dynamic testing. An example of static testing is using tools such as CPP-Check and an example of dynamic security testing would be unit testing.

Risk can be introduced to a system in multiple ways, whether it be from external or inter sources. Regardless of the security threat, the long it takes to fix, the more money a company loses. Lost wages can include loss of user trust, losing internal tools and data, being offline therefore not generating revenue, and the cost to find the security breach and patch it. Some risk is more costly than others, however. For instance, memory protection vulnerabilities have a higher remediation cost than static assertion security vulnerabilities.

Zero trust is a policy that takes a default deny and principle of least privilege approach to security. The policy can be summarized by making sure each person who has access to the system is an authenticated user and that user can only access information and parts of the system that they are authorized to. Some recommendations for the Zero Trust policy is to have a way for external users to gain access using secure login credential, such as email/password or biometrics. For internal users, a proxy or VPN should be used on top of user credentials in order to secure data and internal tools. Additionally, internal users should only have access to data and tools that make it possible to carry out tasks. The less access a user has to systems they do not need/use, the more protected the system is.

Other recommendations for a company would include implementing a DiD policy, encrypting data in its different states, and using accounting to log activity. A DiD policy would be implemented by layering security efforts. The intention behind this is to catch a security threat should it breach one of the other layers. Data should be encrypted whether it be in storage, at flight, or in use. Encrypting data can help protected sensitive information from being accessed if a hacker does gain access to it. Lastly, logging activity on the system can help to identify when abnormal or sudden increase of activity is occurring. This can help to identify and address security threats in a timely manner.