Nicole Penner

8-2 Journal: Portfolio Reflection

CS-405 Secure Coding

Secure coding standards should be adopted and adhered to. This may be to in regards specifically to a programming language and a potential platform. This is one of the secure coding principles that was added in the security policy document. A standard should be created and maintained because they help to be guidelines (Foster, 2020) to prevent as many security vulnerabilities as possible. To ensure your system is as secure as possible, security should not be left until the end. Security should be worked into every step of the creation of the system. This will help to ensure that there are many layers of security, and there are less vulnerabilities in your system. I think that both adhering to coding standards and not leaving security to the end are important and I will be putting them into future programs that I create,

An important step to help with secure coding is the evaluation and assessment of risk and cost benefit of mitigation. To best do this it sometimes takes looking at what the end cost of a security breach related to a certain topic will be, if it will be especially expensive there should be priority placed on preventing it. Just the same as if it is unlikely and will be not as expensive to fix, less priority should be placed on it. Though I think that security should always include ensuring all vulnerabilities are covered, this should be used to place priority on issues that are probable and will have high costs to fix.

Zero trust is the idea that all users need to be authenticated and authorized before gaining access to any part of the application or system. This is an important concept because instead of focusing on just threats that occur outside an organization, threats that come from the inside are also taken seriously. As more workers are remote or hybrid this is important to consider as they access the system. I think this is a useful strategy to adopt because no one is assumed to be trusted, and they are constantly having to be validated to be able to gain access to different parts of the system. I think it is an extremely efficient way to be able to keep a system as secure as possible and I will be sure to use it in the future.

Another topic to discuss is the implementation and recommendation of security policies. When I was creating the security policy document for Green Pace, I had a lot to think about when it came to security policies. I ended up settling on ones that no matter the operating system or language they are important to consider. This may be different depending on what type of program or system you may be using. To help choose the best ones, it may be helpful to read articles and look at literature on security policies. The key security policies that should be followed are: validate input data, heed all compiler warnings, design the system to follow security policies, keep it simple, default deny, adhere to principle of least privilege, sanitize data sent to other systems, practice defense in depth, use quality assurance techniques, and adopt a secure coding standard. These include the idea of zero trust and other factors discussed in this reflection. It is important to follow these and implement them to the best of your abilities in order to keep a system as secure as possible.