Michael Hanlon
CS 405
Secure Coding

Portfolio Reflection

Throughout our research, experimentation, and work through security vulnerabilities, risks, and policies, we have learned a great many things. Secure coding policies should be adopted early and often. The earlier the adoption, the smoother the implementation will be, and the stronger the protection against attacks will be. It will be smoother as there will be less patch work later on, and as attacks can and will be defended against with multiple redundant layers. One example of this can be seen through buffer overflow protection. Developers can limit input to a set amount, verify this amount before sending it off to other systems, and also validate when the input is received by these other systems. This is all in an effort to defend against buffer overflows, but with multiple protections it is ensured that normal functionality will reign supreme. This will lead to ultimately stronger security against all sorts of attacks outlined in the security policy as they will be recognised before they occur and when or if they do occur the team will be prepared to deal with it. While this might seem like a lot of unnecessary defense and money wasted if attacks are not going to occur, it would actually save a bundle compared to remediation costs often. Remediation costs can vary from fixing the system itself after an attack, to steep legal fees if the attack ends up affecting many of your customers/users. Preventing these steep costs early with a little investment early again shows why you should not save security till the end. In addition to all this a zero trust policy should be implemented as well. Zero trust is exactly as it sounds, every user must be authenticated and authorized before they have system access at all. In addition there will be no server privileges as all users should be considered the same amount of threat regardless of if they're on the local company server. Also zero trust invokes data sanitization and validation as any signal or data sent to endpoints will be verified before being allowed in for processing. Zero trust is a strong security measure and also should be implemented from the start, as users and employees might be annoyed if they all the

sudden have to undergo MFA (multi factor authentication), but even if the system does not already have zero trust implemented user annoyance should be considered after attack protection. Overall it's recommended that security is addressed and implemented early and often, protecting against common threats like SQL injection, stack/buffer overflows, memory leaks and corruption, and more. It's important that not only you are protecting against these from the beginning, but also continuously updating and securing in future updates to an ever evolving security landscape.